IoT Device Management

API References

Issue 02

Date 2019-08-28

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

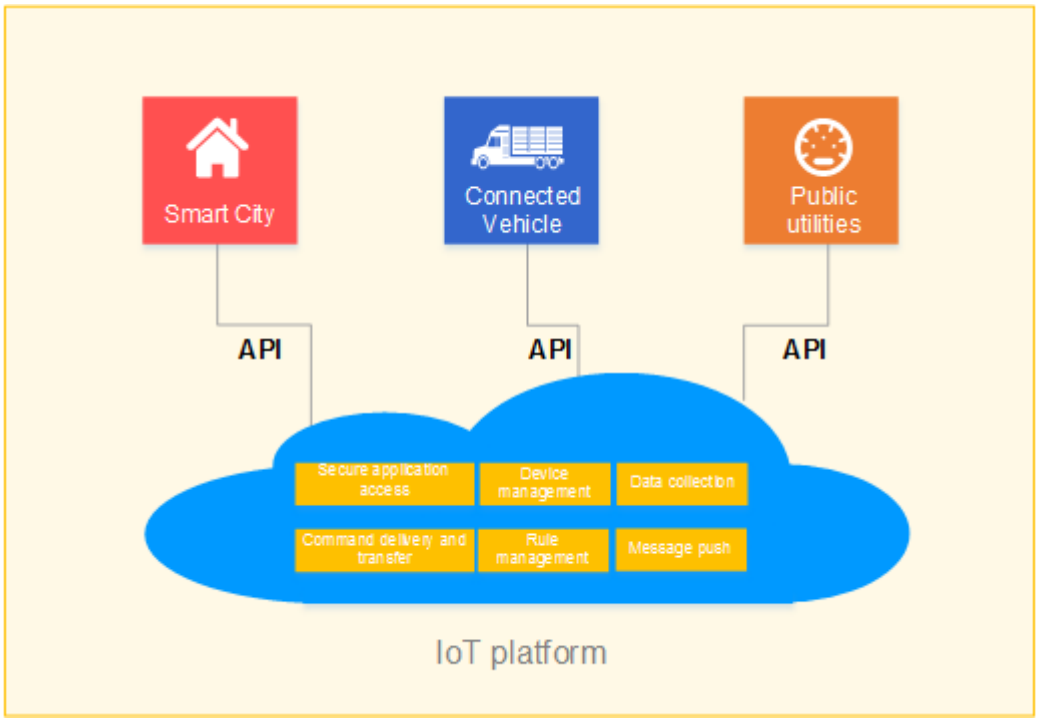
The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Northbound API Reference

1.1 Overview



The IoT platform (IoT platform for short) provides massive APIs for NA developers. By calling IoT platform APIs, developers can develop applications based on various industry devices, such as public utilities and smart home, to manage devices (including adding, deleting, querying, and modifying devices), collect data, deliver commands, and push messages.

1.2 API Usage Statement

- If the API version is evolved or the API URL is modified, APIs of the earlier version can still be used, but their functions are not enhanced. You are advised to use new-version APIs.
- During application development based on the APIs, parameters added in response messages and push messages sent by the IoT platform must be compatible or ignored, preventing application processing exceptions because of incompatible parameters. For parameters that cannot be identified, the IoT platform discards the packets without processing them.
- Before an application calls an API, the device certificate provided by the platform must be preconfigured. You can click **here** to obtain the platform certificate.

1.3 Secure Application Access

After an NA obtains authentication information and connects to the IoT platform, the NA uses the authentication information to call other APIs.

1.3.1 Authentication

Typical Scenario

This API is called by an NA for access authentication when the NA accesses open APIs of the IoT platform for the first time. After the authentication of the NA expires, the NA must call this API to perform authentication again so that the NA can continue to access open APIs of the IoT platform.

API Function

This API is used by an NA to get authenticated before accessing open APIs of the IoT platform for the first time.

Note

The Authentication API is the prerequisite for calling other APIs. **app_key** and **Authorization** must be carried in the request header when northbound APIs, except the Authentication API, are called. The value of **app_key** is the same as that of **appId** in the request. The value of **Authorization** is in the format of **Authorization: Bearer {accessToken}**. The value of **accessToken** is obtained by calling the Authentication API.

If you have obtained the accessToken for multiple times, only the last accessToken is valid, and the previous ones are invalid. Do not obtain the **accessToken** through concurrent attempts.

API Prototype

Method	POST
URL	https://server:port/iocm/app/sec/v1.1.0/login
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(256)	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
secret	Mandatory	String(256)	body	Indicates a secret used to access the IoT platform. It maps to **appId**. The value of this parameter is allocated by the IoT platform when the application is created on the platform.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
scope	String(256)	Indicates the application permission scope, that is, the scope of IoT platform resources that can be accessed using the **accessToken**. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the type of the **accessToken**. This parameter has a fixed value of **Bearer**.
expiresIn	Integer(256)	Indicates the validity period of the **accessToken**. This parameter has a fixed value of 3600 seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used for the Refreshing a Token API. A refreshToken is valid for one month. When the accessToken is about to expire, you can call the Refreshing a Token API to obtain a new one.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/sec/v1.1.0/login
Content-Type: application/x-www-form-urlencoded

appId=******&secret=******
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "scope": "default",
  "tokenType": "Bearer ",
  "expiresIn": "*******",
  "accessToken": "*******",
  "refreshToken": "*******"
}
```

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.3.2 Refreshing a Token

Typical Scenario

An access token obtained by calling the Authentication API has a valid time. When the access token is about to expire, an NA can call this API to obtain a new access token.

API Function

This API is used by an NA to obtain a new access token from the IoT platform when the access token is about to expire.

API Prototype

Method	POST
URL	https://server:port/iocm/app/sec/v1.1.0/refreshToken
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(256)	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
secret	Mandatory	String(256)	body	Indicates a secret used to access the IoT platform. It maps to **appId**. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
refreshToken	Mandatory	String(256)	body	Indicates the refresh token used for obtaining a new **accessToken**. The refreshToken is obtained when you call the Authentication or Refreshing a Token API.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
scope	String(256)	Indicates the applied permission range. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the access token type. This parameter has a fixed value of **Bearer**.
expiresIn	Integer(256)	Indicates the validity period of the **accessToken**. This parameter has a fixed value of 3600 seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used for the Refreshing a Token API. A refreshToken is valid for one month. When the accessToken is about to expire, you can call the Refreshing a Token API to obtain a new one.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/sec/v1.1.0/refreshToken
Content-Type: application/json
Body:
{
  "appId": "******",
  "secret": "******",
  "refreshToken": "******"
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "accessToken": "*******",
  "tokenType": "*******",
  "expiresIn": "*******",
  "refreshToken": "*******",
  "scope": "*******"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4 Device Management

An NA adds a device to the IoT platform and obtains the device ID and verification code. After connecting to the IoT platform, the device establishes a subordinate relationship with the NA.

1.4.1 Registering a Device (Verification Code Mode)

Typical Scenario

Before connecting a device to the IoT platform, an NA must call this API to register the device on the IoT platform and set a unique identification code (such as IMEI) for the device. Then, the device can use the unique identification code to get authenticated and connect to the IoT platform.

This API applies to devices that use LWM2M/CoAP or devices that integrate the AgentLite SDK.

API Function

This API is used by an NA to register a device with the IoT platform. After registration, the device can connect to the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/reg/v1.1.0/deviceCredentials?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceInfo	Optional	**DeviceInfoDTO**	Body	Indicates the device data.

Parameter	Mandatory or Optional	Type	Location	Description
endUserId	Optional	String(256)	Body	Identifies an end user. In the NB-IoT solution, this parameter is set to the IMSI of the device.
imsi	Optional	String(1-64)	Body	Indicates the IMSI of an NB-IoT device.
isSecure	Optional	Boolean	Body	Indicates whether the device is secure. The default value is **false**. In NB-IoT scenarios, if a registered device is an encryption device, this parameter must be set to **true**. ● **true**: The device is secure. ● **false**: The device is not secure.
nodeId	Mandatory	String(256)	Body	Uniquely identifies a device. The value of this parameter must be the same as the device ID reported by the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.
psk	Optional	String(8-32)	Body	If the pre-shared key (PSK) is specified in the request, the IoT platform uses the specified PSK. If the PSK is not specified in the request, the PSK is generated by the IoT platform. The value is a string of characters, including upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.

Parameter	Mandatory or Optional	Type	Location	Description
timeout	Optional	Integer(>=0)	Body	Indicates the validity period of the verification code, in units of seconds. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. The value ranges from 0 to 2147483647. If this parameter is set to **0**, the device verification code is always valid. (The recommended value is **0**.) The default value is **180**. The default value can be configured. For details, contact the IoT platform maintenance personnel.
verifyCode	Optional	String(256)	body	Indicates the verification code of the device. If **verifyCode** is specified in the request, the specified verifyCode is returned in the response. If **verifyCode** is not specified in the request, the verifyCode automatically generated by the IoT platform is returned. Set **verifyCode** when registering a device that has integrated the AgentLite SDK. The value of **verifyCode** must be the same as that of **nodeId**.
productId	Optional	String(256)	Body	Identifies the product to which the device belongs. This parameter is used to associate the product model of the device. Either specify this parameter or the following parameters: **manufacturerId**, **manufacturerName**, **deviceType**, **model**, and **protocolType**.

DeviceInfoDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Optional	String(256)	Body	Uniquely identifies a manufacturer. This parameter, together with **manufacturerName**, **deviceType**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerName**, **deviceType**, **model**, and **protocolType**, or specify **productId**.
manufacturerName	Optional	String(256)	Body	Indicates the manufacturer name. This parameter, together with **manufacturerId**, **deviceType**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **deviceType**, **model**, and **protocolType**, or specify **productId**.
deviceType	Optional	String(256)	Body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**. This parameter, together with **manufacturerId**, **manufacturerName**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **model**, and **protocolType**, or specify **productId**.
model	Mandatory	String(256)	Body	Indicates the device model. This parameter, together with **manufacturerId**, **manufacturerName**, **deviceType**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **deviceType**, and **protocolType**, or specify **productId**.

Parameter	Mandatory or Optional	Type	Location	Description
protocolType	Optional	String(256)	Body	Indicates the protocol used by the device. CoAP and LWM2M are supported. This parameter, together with **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, or specify **productId**.
name	Optional	String	Body	Indicates the device name.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
psk	String(32)	Indicates a random PSK. If the PSK is carried in the request, it is used. Otherwise, the IoT platform generates a random PSK.
timeout	Integer	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.
verifyCode	String(256)	Indicates the verification code of the device. Devices integrated with the AgentLite SDK must use the verification code to complete IoT platform access authentication. If **verifyCode** is specified in the request, the specified verifyCode is returned in the response. If **verifyCode** is not specified in the request, the verifyCode automatically generated by the IoT platform is returned.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/reg/v1.1.0/deviceCredentials?appId=*********
Header:
```



```
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "endUserId": "***********",
  "verifyCode": "****************",
  "nodeId": "****************",
  "deviceInfo": {
    "manufacturerName": "******",
    "manufacturerId": "******",
    "deviceType": "******",
    "model": "******",
    "protocolType": "******",
    "name": "******"
  },
  "psk": "*******************************",
  "timeout": 0
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceId": "*******",
  "verifyCode": "*******",
  "psk": "*********",
  "timeout": 0
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	103028	The license pool resources.	The license resources have been used up.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100416	The device has already been bounded.	The device has been bound to the IoT platform. Recommended handling: Check whether the device is registered.

HTTP Status Code	Error Code	Error Description	Remarks
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application has not been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	600002	The product not existed.	The product does not exist. Recommended handling: Check whether **productId** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100412	The amount of device has reached the limit.	The number of devices under the current application reaches the upper limit. Recommended handling: Check whether the number of devices under the current application reaches the upper limit.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	103026	The license is not exist.	The license does not exist. Recommended handling: An internal license error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.2 Registering a Device (Password Mode)

Typical Scenario

Before connecting a device to the IoT platform, the NA must call this API to register the device on the IoT platform and obtain the device ID and secret. Then, the device can use the device ID and secret to get authenticated and connect to the IoT platform.

This API applies to devices that use MQTT.

API Function

This API is used by an NA to register a device with the IoT platform. After registration, the device can connect to the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/reg/v2.0.0/deviceCredentials?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceInfo	Optional	**DeviceInfo**	Body	Indicates the device data.
endUserId	Optional	String(256)	Body	Identifies an end user. In the NB-IoT solution, this parameter is set to the IMSI of the device.
organization	Optional	String(256)	Body	Indicates the organization to which the device belongs.
region	Optional	String(256)	Body	Indicates the region information about a device.
timezone	Optional	String(256)	Body	Indicates the time zone where the device is located.
mqttConnect	Optional	Boolean	Body	Indicates whether the device uses MQTT for access. Set this parameter to **true** when registering an MQTT device.
productId	Optional	String(256)	Body	Identifies the product to which the device belongs. This parameter is used to associate the product model of the device. Either specify this parameter or the following parameters: **manufacturerId**, **manufacturerName**, **deviceType**, **model**, and **protocolType**.
secret	Optional	String	Body	Indicates the secret of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	Body	Uniquely identifies a device. The value of this parameter must be the same as the device ID reported by the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.
manufacturerId	Optional	String(256)	Body	Uniquely identifies a manufacturer. This parameter, together with **manufacturerName**, **deviceType**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerName**, **deviceType**, **model**, and **protocolType**, or specify **productId**.
manufacturerName	Optional	String(256)	Body	Indicates the manufacturer name. This parameter, together with **manufacturerId**, **deviceType**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **deviceType**, **model**, and **protocolType**, or specify **productId**.

Parameter	Mandatory or Optional	Type	Location	Description
deviceType	Optional	String(256)	Body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**. This parameter, together with **manufacturerId**, **manufacturerName**, **model**, and **protocolType**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **model**, and **protocolType**, or specify **productId**.
model	Mandatory	String(256)	Body	Indicates the device model. This parameter, together with **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, or specify **productId**.
protocolType	Optional	String(256)	Body	Indicates the protocol used by the device. MQTT is supported. This parameter, together with **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, is used to associate the product model of the device. Either specify this parameter, **manufacturerId**, **manufacturerName**, **deviceType**, and **model**, or specify **productId**.
name	Optional	String	Body	Indicates the device name.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
psk	String(32)	Indicates a random PSK generated by the IoT platform.
secret	String(256)	Indicates the device secret, with which the device can get authenticated and connect to the IoT platform. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.

Request Example

```
Method: POST
request:
https://server:port/iocm/app/reg/v2.0.0/deviceCredentials?appId=*********
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "deviceInfo": {
    "nodeId": "******",
    "manufacturerName": "******",
    "manufacturerId": "******",
    "deviceType": "******",
    "model": "******",
    "protocolType": "******",
    "name": "******"
  },
  "endUserId": "***********",
  "organization": "***********",
  "region": "***********",
  "timezone": "***********",
  "mqttConnect": true,
  "productId": "***********",
  "secret": "***********"
}
```

Response Example

```
response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceId": "*******",
  "psk": "*********",
  "secret": "*********"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	600002	The product not existed.	The product does not exist. Recommended handling: Check whether **productId** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100412	The amount of device has reached the limit.	The number of devices under the current application reaches the upper limit. Recommended handling: Check whether the number of devices under the current application reaches the upper limit.
500	103026	The license is not exist.	The license does not exist. Recommended handling: An internal license error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.3 Refreshing a Device Key

Typical Scenario

If the unique identification code of a device that has been registered with the IoT platform changes (for example, a device is replaced), an NA needs to call this API to update the unique identification code of the device and rebind the device.

NOTE

The device password can be updated only when the device is offline.

API Function

This API is used by an NA to update the node ID of a device that has been registered with the IoT platform, and rebind the device with the device ID unchanged.

API Prototype

Method	PUT
URL	https://server:port/iocm/app/reg/v1.1.0/deviceCredentials/{deviceId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String(256)	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Optional	String(256)	body	Uniquely identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. Set this parameter if you need to update the node ID of a device. Otherwise, leave it empty. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.
timeout	Optional	Integer	body	Indicates the validity period for device registration. A device must be bound within the validity period. Otherwise, the registration information will be deleted. The value ranges from 0 to 2147483647. If this parameter is set to **0**, the device verification code is always valid. (The recommended value is **0**.) The default value is **180**. The default value can be configured. For details, contact the IoT platform maintenance personnel. Unit: seconds
verifyCode	Optional	String(256)	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. You are advised to set this parameter to the value of **nodeId**. In the NB-IoT solution, this parameter is mandatory and must be set to the same value as **nodeId**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
verifyCode	String(256)	Indicates the device verification code, with which the device can get authenticated and connect to the IoT platform. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
timeout	Integer	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.

Request Example

```
Method: PUT
Request:
https://server:port/iocm/app/reg/v1.1.0/deviceCredentials/*************?
appId=*************
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "verifyCode": "****************",
  "nodeId": "********",
  "timeout": 300
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "verifyCode": "AE10-12424-12414",
  "timeout": 300
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	100610	Device is not active.	The device has not been activated. Recommended handling: Check whether the device has been connected to the IoT platform and activated.
400	100611	Device is online.	The device is online. Recommended handling: Enable the device to go offline or disconnect the device from the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.4 Modifying Device Information

Typical Scenario

After an NA registers a device with the IoT platform and the basic information about the device changes, the NA can call this API to modify device information on the IoT platform.

API Function

This API is used by an NA to modify the basic information about a device, including the device type, device model, manufacturer, and access protocol.

API Prototype

Method	PUT
URL	https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
customFields	Optional	List<**CustomField**>	Body	User-defined field list. Users can set customized fields.
deviceConfig	Optional	**DeviceConfigDTO**	body	Indicates the device configuration information.
deviceType	Optional	String(1~256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
endUser	Optional	String(1~256)	body	Indicates an end user. If the device is a directly connected device, this parameter is optional. If it is an indirectly connected device, this parameter can be set to **null**.
location	Optional	String(1~1024)	body	Indicates the device location.
productId	Optional	String(256)	Body	Identifies the product to which the device belongs.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Optional	String(1~256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(1~256)	body	Indicates the manufacturer name.
model	Optional	String(1~256)	body	Indicates the device model, which is defined by the manufacturer.
mute	Optional	String(1-256)	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
name	Optional	String(1~256)	body	Indicates the device name.
organization	Optional	String(1~256)	body	Indicates the organization to which the device belongs.
protocolType	Optional	String(1~256)	body	Indicates the protocol used by the device. CoAP, LWM2M, and MQTT are supported.
region	Optional	String(1~256)	body	Indicates the region information about a device.
timezone	Optional	String(1~256)	body	Indicates the time zone of the device, for example, Asia/Shanghai and America/New_York.
imsi	Optional	String(64)	Body	Indicates the IMSI of an NB-IoT device.
ip	Optional	String(128)	Body	Indicates the device IP address.
isSecure	Optional	Boolean	body	Indicates whether the device is secure. The default value is **false**. In NB-IoT scenarios, if a registered device is an encryption device, this parameter must be set to **true**. ● **true**: The device is secure. ● **false**: The device is not secure.

Parameter	Mandatory or Optional	Type	Location	Description
psk	Optional	String(8-32)	body	Indicates the pre-shared key (PSK). The value is a string of characters, including upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.
tags	Optional	List<Tag2>	Body	Indicates the tag of a device.

CustomField structure

Parameter	Mandatory or Optional	Type	Location	Description
fieldName	Optional	String(256)	Body	Indicates the field name.
fieldType	Optional	String(256)	Body	Indicates the field type.
fieldValue	Optional	Object	Body	Indicates the field value.

DeviceConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataConfig	Optional	DataConfigDTO	Indicates the data configuration information.

DataConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataAgingTime	Optional	Integer[0,90]	Indicates the data aging time. The value range is 0 - 90. Unit: day.

Tag2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.
tagType	Optional	Integer	body	Label Type

Response Parameters

Status Code: 204 No Content

Request Example

```
Method: PUT
Request:
https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?appId={appId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "name": "****",
  "endUser": "****",
  "mute": "****",
  "deviceConfig": {
    "dataConfig": {
      "dataAgingTime": 30
    }
  }
}
```

Response Example

```
Response:
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100440	The isSecure is invalid.	The value of **isSecure** is incorrect.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	500004	The amount of frozen devices has reached the limit.	The number of frozen devices has reached the upper limit.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.5 Deleting a Device

Typical Scenario

If a device that has been registered on the IoT platform does not need to connect to the IoT platform, an NA can call this API to delete the device. If the device needs to connect to the IoT platform again, register it again.

API Function

This API is used by an NA to delete a registered device from the IoT platform.

API Prototype

Method	DELETE

URL	https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?appId={appId}&cascade={cascade}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
cascade	Optional	Boolean	query	This parameter is valid only when the device is connected to indirectly-connected devices. The default value is **true**. ● **true**: The directly connected device and the indirectly-connected devices connected to it are deleted. ● **false**: The directly connected device is deleted but the indirectly-connected devices connected to it are not deleted. In addition, the attribute of the indirectly-connected devices is changed to directly-connected.

Response Parameters

Status Code: 204 No Content

Request Example

```
Method: DELETE
Request:
https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?
appId={appId}&cascade={cascade}
Header:
app_key: ******
Authorization: Bearer ******
```

Response Example

```
Response:
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.6 Deleting a Child Device

If a child device does not need to connect to the IoT platform, an NA can use this API to delete the device from the gateway.

This function is implemented by delivering a command to the gateway through the **Calling Device Services** API. The command is negotiated by the NA and the gateway and is defined in the device profile file.

After receiving the command from the IoT platform, the gateway starts the internal service process to delete the child device.

1.4.7 Querying Device Activation Status

Typical Scenario

After an NA registers a device on the IoT platform, the activation status of the device is **false** before the device connects to the IoT platform for the first time. When the device connects to the IoT platform for the first time, the activation status of the device is **true** regardless of whether the device is online, offline, or abnormal. The NA can call this API to query the activation status of the device to check whether the device has connected to the IoT platform.

API Function

This API is used by an NA to query the activation status of a device on the IoT platform to determine whether the device has connected to the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/reg/v1.1.0/deviceCredentials/{deviceId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
activated	Boolean	Indicates whether the device is activated through a verification code. ● **true**: The device is activated. ● **false**: The device is not activated.
name	String(256)	Indicates the device name.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/reg/v1.1.0/deviceCredentials/{deviceId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceId": "*******",
  "activated": "*******",
  "name": "*******"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.8 Modifying Device Shadow Information

Typical Scenario

The IoT platform supports the creation of device shadows. Device shadows store the latest service property data reported by devices and service property configurations delivered by an NA. (Service properties are defined in the device profile file.) If the device is offline or abnormal, the NA cannot deliver configuration to the device by delivering commands. In this case, the NA can set the configuration to the device shadow. When the device goes online again, the device shadow delivers the configuration to the device. The NA can call this API to modify the configuration information to be delivered to the device on the device shadow.

Each device has only one device shadow, which contains desired and reported sections.

- The desired section stores the configurations of device service properties. If a device is online, the configurations in the desired section are delivered to the device immediately. Otherwise, the configurations in the desired section are delivered to the device when the device goes online.
- The reported section stores the latest service property data reported by devices. When a device reports data, the IoT platform synchronizes the data to the reported section of the device shadow.

API Function

This API is used by an NA to modify the configuration information in the desired section of the device shadow. When the device goes online, the configuration information will be delivered to the device.

Note

Only LWM2M-based devices support the device shadow function, and only properties defined by LWM2M can be modified. User-defined properties cannot be modified.

API Prototype

Method	PUT
URL	https://server:port/iocm/app/shadow/v1.5.0/devices/{deviceId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
serviceDesireds	Mandatory	List<**ServiceDesiredDTO**>	body	Indicates the configuration or status to be modified.

ServiceDesiredDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-256)	body	Identifies a service.
desired	Mandatory	ObjectNode	body	Indicates the service attribute configuration of a device.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: PUT
Request:
https://server:port/iocm/app/shadow/v1.5.0/devices/devices/{deviceId}?
appId={appId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "serviceDesireds": [
    {
      "serviceId": "Temperature",
      "desired": {
        "targetTemperature": 35
      }
    }
  ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100425	The special deviceCapability is not exist.	The device template does not exist. Recommended handling: Check whether the device template has been uploaded to the IoT platform.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.

HTTP Status Code	Error Code	Error Description	Remarks
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100443	The property is forbidden to write.	The device attributes cannot be written.
403	10100 09	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	10100 05	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.4.9 Querying Device Shadow Information

Typical Scenario

When a device is in the offline or abnormal state, an NA cannot deliver configuration information to the device by sending a command. In this case, the NA can deliver the configuration information to the device shadow. When the device goes online, the device shadow will deliver the configuration information to the device. The NA can call this API to check the device configuration information and the latest data reported by the device on the device shadow.

API Function

This API is used by an NA to query the device shadow information of a device, including the device configuration information (in the desired section) and the latest data reported by the device (in the reported section).

Note

Only LWM2M-based devices support the device shadow function, and only properties defined by LWM2M can be modified. User-defined properties cannot be modified.

API Prototype

Method	GET
URL	https://server:port/iocm/app/shadow/v1.5.0/devices/{deviceId}?appId={appId}

Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String(36)	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
deviceId	String(36)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	String(36)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.

Parameter	Type	Description
nodeType	Enum	Indicates the device type.
createTime	String(256)	Indicates the device creation time.
lastModifiedTime	String(256)	Indicates the last modification time.
deviceInfo	**DeviceInfo**	Indicates detailed information of the device.
services	List<**DeviceServiceB**>	Indicates the service attribute information of the device.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Uniquely identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model.
swVersion	String(256)	Indicates the software version of the device.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol used by the device.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.

Parameter	Type	Description
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.
statusDetail	String(256)	Indicates the device status details, which vary according to the value of status. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.

Parameter	Type	Description
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

DeviceServiceB structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
reportedProps	ObjectNode	Indicates the latest data reported by the device.
desiredProps	ObjectNode	Indicates the configuration information delivered to the device.
eventTime	String(256)	Indicates the time when an event occurs.
serviceType	String(256)	Indicates the service type.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/shadow/v1.5.0/devices/{deviceId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceId": "******",
  "gatewayId": "******",
  "nodeType": "******",
  "createTime": "******",
  "lastModifiedTime": "******",
  "deviceInfo": "******""services": [
    {
      "serviceId": "******",
      "reportedProps": "******",
      "desiredProps": "******",
      "eventTime": "******",
      "serviceType": "******"
    },
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.5 Data Collection

The IoT platform allows NAs to query basic information about a device and view historical data reported by devices by hour, day, or month.

1.5.1 Querying Information About a Device

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of a device that has been registered on the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information of a specified device based on the device ID on the IoT platform, such as configuration, status and service attributes.

API Prototype

Method	GET
URL	https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?appId={appId}&select={select}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.

Parameter	Mandatory or Optional	Type	Location	Description
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
select	Optional	String	query	Indicates the query condition. The value can be **IMSI**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.

Parameter	Type	Description
deviceInfo	**DeviceInfoQueryDTO**	Indicates the device data.
services	List<**DeviceService**>	Indicates the service list.
alarmInfo	**AlarmInfoDTO**	Indicates the device alarm details.

DeviceInfoQueryDTO structure

Parameter	Type	Description
nodeId	String(256)	Uniquely identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model.
swVersion	String(256)	Indicates the software version of the device.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
imsi	String	Indicates the IMSI of an NB-IoT device.
protocolType	String(256)	Indicates the protocol used by the device.
radiusIp	String	Indicates the RADIUS address.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.

Parameter	Type	Description
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.
statusDetail	String(256)	Indicates the device status details, which vary according to the value of status. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.

Parameter	Type	Description
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
serviceInfo	**ServiceInfo**	Indicates the service information.
data	ObjectNode(2097152)	Indicates an attribute-value pair (AVP).
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the list of shielded device commands.

AlarmInfoDTO structure

Parameter	Type	Description
alarmSeverity	String	Indicates the alarm severity.
alarmStatus	Boolean	Indicates the alarm status.
alarmTime	String	Indicates the time when the alarm is reported.

Request Example

```
Method: GET
Request:
```

```
https://server:port/iocm/app/dm/v1.4.0/devices/{deviceId}?select=imsi
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceId": "xxxxx",
  "gatewayId": "xxxxx",
  "nodeType": "xxxxx",
  "deviceInfo": {
    "nodeId": "123456",
    "name": "Sensor_12",
    "manufacturerName": "wulian",
    "deviceType": "gateway",
    "model": "90",
    "mac": "C7EA1904004B1204",
    "swVersion": "th",
    "fwVersion": "seu",
    "hwVersion": "sru",
    "protocolType": "zigbee",
    "description": "smockdetector",
    "imsi": "xxxxx"
  },
  "services": [
    {
      "serviceType": "air_conditioner",
      "serviceId": "1",
      "data"{
        "battery_low": 1
      }
    },
    {
      "serviceType": "air_conditioner",
      "serviceId": "jkh",
      "data": {
        "battery_low": "jhj"
      }
    }
  ]
}
```

Error Codes

HTTP Status Code	Error Codes	Error Description	Remarks
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.

HTTP Status Code	Error Codes	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Codes	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.5.2 Querying Device Information in Batches

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of multiple devices that have been registered on the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information (such as configuration, status and service attributes) of multiple devices based on specified conditions on the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/dm/v1.4.0/devices?appId={appId}&deviceType={deviceType}&endTime={endTime}&gatewayId={gatewayId}&location={location}&name={name}&nodeType={nodeType}&pageNo={pageNo}&pageSize={pageSize}&select={select}&sort={sort}&startTime={startTime}&status={status}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
gatewayId	Optional	String	query	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. **gatewayId** and **pageNo** cannot be both left empty.
nodeType	Optional	String	query	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
deviceType	Optional	String	query	Indicates the device type.
location	Optional	String	query	Indicates the device location.
name	Optional	String	query	Indicates the device name.
pageNo	Optional	Integer	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried. **gatewayId** and **pageNo** cannot be both left empty.

Parameter	Mandatory or Optional	Type	Location	Description
pageSize	Optional	Integer	query	Indicates the number of records to be displayed on each page. Pagination is implemented based on the value of this parameter. The value is an integer ranging from 1 to 250. The default value is **25**.
status	Optional	String	query	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.
startTime	Optional	String	query	Indicates the start time. Records with the device registration time later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Records with the device registration time earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
sort	Optional	String	query	Indicates the sorting mode of queried records. ● **ASC**: Records are sorted in ascending order of device registration time. ● **DESC**: Records are sorted in descending order of device registration time. The default value is **DESC**.
select	Optional	String	query	Indicates the record to be returned. The value can be **IMSI**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
devices	List<**QueryDeviceDTO4Cloud2NA**>	Indicates the device list.

QueryDeviceDTO4Cloud2NA structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.
deviceInfo	**DeviceInfoQueryDTO**	Indicates the device data.
services	List<**DeviceService**>	Indicates the service list.
alarmInfo	**AlarmInfoDTO**	Indicates the device alarm details.

DeviceInfoQueryDTO structure

Parameter	Type	Description
nodeId	String(256)	Uniquely identifies a device.
name	String(256)	Indicates the device name.

Parameter	Type	Description
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model.
swVersion	String(256)	Indicates the software version of the device.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
imsi	String	Indicates the IMSI of an NB-IoT device.
protocolType	String(256)	Indicates the protocol used by the device.
radiusIp	String	Indicates the RADIUS address.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.

Parameter	Type	Description
statusDetail	String(256)	Indicates the device status details, which vary according to the value of status. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
serviceInfo	**ServiceInfo**	Indicates the service information.
data	ObjectNode(2097152)	Indicates an attribute value pair.
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the list of shielded device commands.

AlarmInfoDTO structure

Parameter	Type	Description
alarmSeverity	String	Indicates the alarm severity.
alarmStatus	Boolean	Indicates the alarm status.
alarmTime	String	Indicates the time when the alarm is reported.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/dm/v1.4.0/devices?gatewayId={gatewayId}&select=imsi
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "totalCount":"****",
    "pageNo":"*****",
    "pageSize":"*****",
    "devices":[
        {
            "deviceId":"xxxxx",
```

```
            "gatewayId":"xxxxx",
            "nodeType":"xxxxx",
            "deviceInfo":{
                "nodeId": "123456",
                "name":"Sensor_12",
                "manufacturerName":"wulian",
                "deviceType":"gateway",
                "model":"90",
                "mac":"C7EA1904004B1204",
                "swVersion":"th",
                "fwVersion":"seu",
                "hwVersion":"sru",
                "protocolType":"zigbee",
                "description":"smockdetector",
                "imsi":"xxxxx"
            },
            "services":[
                {
                    "serviceType":"air_conditioner",
                    "serviceId":"1",
                    "data":{
                        "battery_low":"1"
                    }
                },
                {
                    "serviceType":"air_conditioner",
                    "serviceId":"jkh",
                    "data":{
                        "battery_low":"jhj"
                    }
                }
            ]
        },
        {
            "deviceId":"xxxxx",
            "gatewayId":"xxxxx",
            "nodeType":"xxxxx",
            "deviceInfo":{
                "nodeId": "223456",
                "name":"Sensor_12",
                "manufacturerName":"wulian",
                "type":"90",
                "model":" 90",
                "mac":"C7EA1904004B1204",
                "swVersion":"…",
                "fwVersion":"…",
                "hwVersion":"…",
                "protocolType":"zigbee",
                "description":"smockdetector",
                "imsi":"xxxxx"
            },
            "services":[
                {
                    "serviceType":"air_conditioner",
                    "serviceId":"1",
                    "data":{
                        "battery_low":"1"
                    }
                },
                {
                    "serviceType":"air_conditioner",
                    "serviceId":"1",
                    "data":{
                        "battery_low":"1"
                    }
                }
            ]
        }
```



```
    ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100218	The gatewayId and pageNo cannot be both null.	The **gatewayId** and **pageNo** parameters cannot be null at the same time. Recommended handling: Check whether **gatewayId** or **pageNo** is set.
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.5.3 Querying Historical Device Data

Typical Scenario

The IoT platform receives and saves service data reported by devices during daily operation. If an NA needs to view the historical data reported by a device to the IoT platform, the NA can call this API to obtain the data.

API Function

This API is used by an NA to query historical data reported by a specified device to the IoT platform based on the device ID.

API Prototype

Method	GET
URL	https://server:port/iocm/app/data/v1.2.0/deviceDataHistory?deviceId={deviceId}&gatewayId={gatewayId}&appId={appId}&pageNo={pageNo}&pageSize={pageSize}&startTime={startTime}&endTime={endTime}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceId	Mandatory	String	query	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Mandatory	String	query	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
pageNo	Optional	Integer	query	Indicates the page number. The default value is **0**, indicating that the first page is queried. You are advised to set **pageNo** and **pageSize** to query historical device data.

Parameter	Mandatory or Optional	Type	Location	Description
pageSize	Optional	Integer	query	Indicates the number of records to be displayed on each page. The default value is **1** and the maximum value is **2000**. Set this parameter based on the average size of data reported by devices. Ensure that the data package of a single query does not exceed 16 MB, and set the timeout period based on the package size of a single query and network bandwidth.
startTime	Optional	String	query	Indicates the start time. Historical shadow data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Historical shadow data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
deviceDataHistoryDTOs	List<**DeviceDataHistoryDTO**>	Indicates the list of historical device data.

DeviceDataHistoryDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.

Parameter	Type	Description
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
appId	String(256)	Identifies the application to which the device belongs.
data	JsonObject	Indicates the data reported by the device.
timestamp	String(256)	Indicates the timestamp when the data is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/data/v1.2.0/deviceDataHistory?
deviceId={deviceId}&gatewayId={gatewayId}&appId={appId}&pageNo={pageNo}&pageSize={
pageSize}&startTime={startTime}&endTime={endTime}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "totalCount":"****",
    "pageNo":"*****",
    "pageSize":"*****",
    "deviceDataHistoryDTOs":[
        {
            "serviceId":"*****",
            "deviceId":"*****",
            "gatewayId":"*****",
            "appId":"*****",
            "data":"*****",
            "timestamp":"*****"
        }
    ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description. For example, check whether the value of **pageSize** exceeds 2000.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.5.4 Querying Historical Device Shadow Data

Typical Scenario

When an NA modifies the configuration of a device shadow by calling the Modifying Device Shadow Information API, the IoT platform saves the modification record. If the NA needs to view historical configuration records of the device shadow, the NA can call this API to obtain the records.

API Function

This API is used by an NA to query historical configuration data about a device shadow based on the device ID.

API Prototype

Method	GET
URL	https://server:port/iocm/app/shadow/v1.5.0/deviceDesiredHistory?deviceId={deviceId}&gatewayId={gatewayId}&appId={appId}&serviceId={serviceId}&property={property}&pageNo={pageNo}&pageSize={pageSize}&startTime={startTime}&endTime={endTime}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceId	Mandatory	String	query	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Mandatory	String	query	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
serviceId	Optional	String	query	Identifies a service.
property	Optional	String	query	Indicates the service attribute.
pageNo	Optional	Integer	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records to be displayed on each page. Pagination is implemented based on the value of this parameter. The value is an integer ranging from 1 to 250. The default value is **25**.

Parameter	Mandatory or Optional	Type	Location	Description
startTime	Optional	String	query	Indicates the start time. Historical shadow data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Historical shadow data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
DeviceDesiredHistoryDTO	List<**DeviceDesiredHistoryDTO**>	Indicates the list of historical device shadow configuration data.

DeviceDesiredHistoryDTO structure:

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
appId	String(256)	Identifies the application to which the device belongs.

Parameter	Type	Description
desired	JsonObject	Indicates the configuration information delivered to the device.
timestamp	String(256)	Indicates the timestamp when the data is configured. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/shadow/v1.5.0/deviceDesiredHistory?
deviceId={deviceId}&gatewayId={gatewayId}&appId={appId}&serviceId={serviceId}&prop
erty={property}&pageNo={pageNo}&pageSize={pageSize}&startTime={startTime}&endTime
={endTime}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "totalCount":"****",
    "pageNo":"*****",
    "pageSize":"*****",
    "DeviceDesiredHistoryDTO":[
        {
            "serviceId":"*****",
            "deviceId":"*****",
            "gatewayId":"*****",
            "appId":"*****",
            "desired":"*****",
            "timestamp":"*****"
        }
    ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.5.5 Querying Service Capabilities of a Device

Typical Scenario

If an NA needs to know which service attributes can be reported by a device and which commands can be delivered to the device, the NA can call this API to query the device service capabilities defined in the profile file of the device on the IoT platform.

API Function

This API is used by an NA to query device service capabilities, such as service attributes and device commands.

API Prototype

Method	GET
URL	https://server:port/iocm/app/data/v1.1.0/deviceCapabilities?appId={appId}&gatewayId={gatewayId}&deviceId={deviceId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
gatewayId	Optional	String	query	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceId	Optional	String	query	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.

Response Parameters

Parameter	Type	Description
deviceCapabilities	List<**DeviceCapabilityDTO**>	Indicates the query result list.

DeviceCapabilityDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
serviceCapabilities	List<**ServiceCapabilityDTO**>	Indicates the list of service capabilities of a device.

ServiceCapabilityDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
option	String(256)	Indicates a service option.
description	String(10240)	Indicates the service description.
commands	List<**ServiceCommand**>	Indicates the list of supported commands.
properties	List<**ServiceProperty**>	Indicates the list of supported properties.

ServiceCommand structure

Parameter	Type	Description
commandName	String(256)	Indicates the command name.
paras	List<**ServiceCommandPara**>	Indicates the attribute list.
responses	List<**ServiceCommandResponse**>	Specifies the list of responses.

ServiceCommandPara structure

Parameter	Type	Description
paraName	String(256)	Indicates the parameter name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.

Parameter	Type	Description
min	String	Indicates the minimum value of the attribute.
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

ServiceCommandResponse structure

Parameter	Type	Description
responseName	String(256)	Indicates the response name.
paras	List<**ServiceCommandPara**>	Indicates the attribute list.

ServiceProperty structure

Parameter	Type	Description
propertyName	String(256)	Indicates the attribute name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.
min	String	Indicates the minimum value of the attribute.
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
method	String(256)	Indicates the access method. The values are as follows: ● **R**: Readable ● **W**: Writable ● **E**: Observable
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/data/v1.1.0/deviceCapabilities?
appId={appId}&gatewayId={gatewayId}&deviceId={deviceId}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "deviceCapabilities":[
        {
            "deviceId":"****",
            "serviceCapabilities":[
                {
                    "serviceId":"****",
                    "serviceType":"****",
                    "option":"****",
                    "description":"****",
                    "commands":[
                        {
                            "commandName":"*****",
                            "paras":[
                                {
                                    "paraName":"*****",
                                    "dataType":"*****",
                                    "required":"Ture",
                                    "min":"*****",
                                    "max":"*****",
                                    "step":"*****",
                                    "maxLength":1111111,
                                    "unit":"*****",
                                    "enumList":[
                                        {

                                        }
                                    ]
                                }
                            ],
                            "responses":[
                                {
                                    "responseName":"***",
                                    "paras":[
                                        {
                                            "paraName":"*****",
                                            "dataType":"*****",
                                            "required":"Ture",
                                            "min":"*****",
                                            "max":"*****",
                                            "step":"*****",
                                            "maxLength":1111111,
                                            "unit":"*****",
                                            "enumList":[
                                                {

                                                }
                                            ]
                                        }
                                    ]
                                }
                            ],
```



```
                            "properties":[
                                {
                                    "propertyName":"*****",
                                    "dataType":"*****",
                                    "required":"Ture",
                                    "min":"*****",
                                    "max":"*****",
                                    "step":"*****",
                                    "maxLength":1111111,
                                    "method":"*****",
                                    "unit":"*****",
                                    "enumList":[
                                        {

                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.6 Subscription Management

The IoT platform allows NAs to subscribe to device data. If the subscribed device data changes, the IoT platform pushes change notifications to NAs. The subscription management APIs must be used together with the APIs used for **message push**.

1.6.1 Subscribing to Service Data of the IoT Platform

Typical Scenario

An NA can subscribe to service data of a device on the IoT platform. When the service data changes (for example, the device is registered, the device reports data or the device status changes), the IoT platform can push change notifications to the NA.

API Function

This API is used by an NA to subscribe to service change notifications on the IoT platform. When the device status or data changes, the IoT platform pushes notifications to the NA.

API Prototype

Method	POST
URL	https://server:port/iocm/app/sub/v1.2.0/subscriptions?ownerFlag={ownerFlag}

Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
ownerFlag	Optional	String(256)	query	Indicates the owner flag of the callback URL. ● **false**: The callback URL owner is an authorizing application. ● **true**: The callback URL owner is an authorized application.
appId	Optional	String(256)	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type based on which an NA can receive notifications pushed by the IoT platform. ● **deviceAdded**: device addition; sending such a notification after subscription ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceCapabilitiesChanged**: device service capability change; sending such a notification after subscription ● **deviceCapabilitiesAdded**: device service capability added; sending such a notification after subscription ● **deviceCapabilitiesDeleted**: device service capability deleted; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription (This function applies only to devices that use the MQTT protocol for access.) ● **commandRsp**: command response; sending such a notification after subscription. (This function applies only to devices that use the MQTT protocol for access.) ● **ruleEvent**: rule event; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow

Parameter	Mandatory or Optional	Type	Location	Description
				modification status change; sending such a notification after subscription
callbackUrl	Mandatory	String(1024)	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. HTTPS or HTTP callback URL can be used. HTTPS is recommended. The callback URL can be a combination of an IP address or a domain name and a port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**.
channel	Optional	String(32)	Body	Indicates the transmission channel. For the MQTT client, the value is **MQTT**. In other cases, the value is **HTTP**.

Response Parameters

Status Code: 201 Created

Parameter	Type	Description
subscriptionId	String	Uniquely identifies a subscription. The value of this parameter is allocated by the IoT platform during subscription creation.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.
clientIds	List<String>	Identifies an MQTT client. The value is that returned when MQTT is subscribed to.

Request Example

```
Method: POST
request: https://server:port/iocm/app/sub/v1.2.0/subscriptions
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
Body:
{
  "appId": "*******",
```



```
  "notifyType": "deviceInfoChanged",
  "callbackUrl": "https://***********"
}
```

Response Example

```
Response:
Status Code: 201 Created
Content-Type: application/json
Body:
{
  "subscriptionId": "*******",
  "notifyType": "*******",
  "callbackUrl": "https://***********"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	The request callbackurl is illegal.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
409	100227	The resource is conflicted.	A resource conflict occurs. The notification type has been subscribed to. Recommended handling: Check whether the notification type has been subscribed.

1.6.2 Subscribing to Management Data of the IoT Platform

Typical Scenario

An NA can subscribe to management data of a device on the IoT platform. When operations are performed on the device (for example, device upgrade), the IoT platform notifies the NA of the operating status or results.

API Function

This API is used by an NA to subscribe to device upgrade notifications on the IoT platform. When the device is upgraded, the IoT platform sends a notification to the NA.

API Prototype

Method	POST
URL	https://server:port/iodm/app/sub/v1.1.0/subscribe
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. ● **swUpgradeStateChangeNotify**: software upgrade status change notification; sending such a notification after subscription ● **swUpgradeResultNotify**: software upgrade result notification; sending such a notification after subscription ● **fwUpgradeStateChangeNotify**: hardware upgrade status change notification; sending such a notification after subscription ● **fwUpgradeResultNotify**: hardware upgrade result notification; sending such a notification after subscription
callbackUrl	Mandatory	String	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. HTTPS or HTTP callback URL can be used. HTTPS is recommended. The callback URL can be a combination of an IP address or a domain name and a port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
request: https://server:port/iodm/app/sub/v1.1.0/subscribe
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
Body:
{
  "notifyType": "swUpgradeStateChangeNotify",
  "callbackUrl": "http://******"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	Internal server error.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
400	100228	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100229	Get AppKey from header failed.	Failed to obtain the AppKey from the message header.
500	100244	register out route fail.	Failed to register the route. Recommendation: Contact the IoT platform maintenance personnel.

1.6.3 Querying a Subscription

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query configuration information about a subscription.

API Function

This API is used by an NA to query the configuration information about a subscription by subscription ID on the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/sub/v1.2.0/subscriptions/{subscriptionId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
subscriptionId	Mandatory	String(256)	path	Uniquely identifies a subscription. The value of this parameter is allocated by the IoT platform during subscription creation.
appId	Optional	String(256)	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
subscriptionId	String	Uniquely identifies a subscription. The value of this parameter is allocated by the IoT platform during subscription creation.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Request Example

```
Method: GET
request: https://server:port/iocm/app/sub/v1.2.0/subscriptions/{subscriptionId}?
appId={appId}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "subscriptionId": "*******",
  "notifyType": "*******",
  "callbackUrl": "*******"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

1.6.4 Querying Subscription in Batches

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query all subscription configurations of the current application or of a specified subscription type.

API Function

This API is used to query all subscription information of the current application or of a specified subscription type.

API Prototype

Method	GET
URL	https://server:port/iocm/app/sub/v1.2.0/subscriptions?appId={appId}&notifyType={notifyType}&pageNo={pageNo}&pageSize={pageSize}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String(256)	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **deviceAdded**: device addition; sending such a notification after subscription ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceCapabilitiesChanged**: device service capability change; sending such a notification after subscription ● **deviceCapabilitiesAdded**: device service capability added; sending such a notification after subscription ● **deviceCapabilitiesDeleted**: device service capability deleted; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription ● **commandRsp**: command response; sending such a notification after subscription ● **ruleEvent**: rule event; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change; sending such a notification after subscription ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification

Parameter	Mandatory or Optional	Type	Location	Description
				• **fwUpgradeStateChangeNotify**: firmware upgrade status change notification • **fwUpgradeResultNotify**: firmware upgrade result notification
pageNo	Optional	Integer	query	Indicates the page number. • If the value is **null**, pagination query is not performed. • If the value is an integer greater than or equal to 0, pagination query is performed. • If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **10**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	Long(>=0)	Indicates the total number of records.
pageNo	Long(>=0)	Indicates the page number.
pageSize	Long(>=0)	Indicates the number of records on each page.
subscriptions	List<**QuerySubscriptionDTOCloud2NA**>	Indicates the subscription information list.

QuerySubscriptionDTOCloud2NA structure

Parameter	Type	Description
subscriptionId	String	Uniquely identifies a subscription. The value of this parameter is allocated by the IoT platform during subscription creation.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Request Example

```
Method: GET
request:https://server:port/iocm/app/sub/v1.2.0/subscriptions?
appId={appId}&notifyType={notifyType}&pageNo={pageNo}&pageSize={pageSize}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "pageNo": 0,
  "pageSize": 100,
  "totalCount": 10,
  "subscriptions": [
    {
      "subscriptionId": "*******",
      "notifyType": "*******",
      "callbackUrl": "*******"
    }
  ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100224	The resource exceeds 1000, please refinement query conditions.	The number of resources exceeds 1000. Recommended handling: Narrow down the filter criteria.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

1.6.5 Deleting a Subscription

Typical Scenario

If an NA does not need to receive a subscription notification message pushed by the IoT platform, the NA can call this API to delete the specified subscription configuration to cancel the subscription.

API Function

This API is used by an NA to delete the configuration information about a subscription by subscription ID on the IoT platform.

API Prototype

Method	DELETE
URL	https://server:port/iocm/app/sub/v1.2.0/subscriptions/{subscriptionId}?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.

Parameter	Mandatory or Optional	Type	Location	Description
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
subscriptionId	Mandatory	String(256)	path	Uniquely identifies a subscription. The value of this parameter is allocated by the IoT platform during subscription creation.
appId	Optional	String(256)	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 204 No Content

Request Example

```
Method: DELETE
request: https://server:port/iocm/app/sub/v1.2.0/subscriptions/{subscriptionId}?
appId={appId}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **subscriptionId** is correct.

1.6.6 Deleting Subscriptions in Batches

Typical Scenario

If an NA does not need to receive subscription notification messages pushed by the IoT platform or a specified type of subscription notification messages, the NA can call this API to delete subscription configurations in batches to cancel the subscriptions.

API Function

This API is used to delete all subscriptions, subscriptions of a specified subscription type, or subscriptions of a specified callback URL in batches.

API Prototype

Method	DELETE
URL	https://server:port/iocm/app/sub/v1.2.0/subscriptions?appId={appId}&notifyType={notifyType}&callbackUrl={callbackUrl}&channel={channel}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String(256)	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **deviceAdded**: device addition; sending such a notification after subscription ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceCapabilitiesChanged**: device service capability change; sending such a notification after subscription ● **deviceCapabilitiesAdded**: device service capability added; sending such a notification after subscription ● **deviceCapabilitiesDeleted**: device service capability deleted; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription ● **commandRsp**: command response; sending such a notification after subscription ● **ruleEvent**: rule event; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change; sending such a notification after subscription ● **swUpgradeStateChangeNotify**: software upgrade status change notification

Parameter	Mandatory or Optional	Type	Location	Description
				● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification
callbackUrl	Optional	String(256)	query	Indicates the callback URL of the subscription.
channel	Optional	String(32)	query	Indicates the transmission channel. ● If this parameter is left unspecified, the IoT platform deletes subscriptions with **channel** set to **HTTP**. ● If this parameter is set to **MQTTS**, the IoT platform deletes only subscriptions with **channel** set to **MQTTS**. ● If this parameter is set to **HTTP**, the IoT platform deletes only subscriptions with **channel** set to **HTTP**. ● If this parameter is set to other values, subscriptions are not deleted.

Response Parameters

Status Code: 204 No Content

Request Example

```
Method: DELETE
request: https://server:port/iocm/app/sub/v1.2.0/subscriptions?
appId={appId}&notifyType={notifyType}&callbackUrl={callbackUrl}&channel={channel}
Header:
app_key: ******
Authorization: Bearer *****
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 204 No Content
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **notifyType** is correct.

1.6.7 Push Notification

NAs can subscribe to device information from the IoT platform. When the device information changes, the IoT platform pushes change notifications to the NAs. Then, the NAs distribute messages based on the notification type. The message push APIs must be used together with the APIs used for **subscription management**.

1.6.7.1 Pushing Device Registration Notifications

Typical Scenario

After an NA subscribes to device registration notifications (the notification type is **deviceAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when the NA registers a device on the IoT platform by calling the Registering Devices API.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device registration notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceAdded**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Optional	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Mandatory	String	body	Indicates the device type. ● ENDPOINT ● GATEWAY ● UNKNOWN
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
protocolType	Optional	String(256)	body	Indicates the protocol used by the device.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.
statusDetail	Optional	String(256)	body	Indicates the device status details, which vary according to the value of **status**. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.

Parameter	Mandatory or Optional	Type	Location	Description
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
```



```
Content-Type: application/json
Body:
{
  "notifyType": "deviceAdded",
  "deviceId": "*****",
  "gatewayId": "*****",
  "nodeType": "GATEWAY",
  "deviceInfo": {
    "nodeId": "******",
    "name": null,
    "description": null,
    "manufacturerId": null,
    "manufacturerName": null,
    "mac": null,
    "location": null,
    "deviceType": null,
    "model": null,
    "swVersion": null,
    "fwVersion": null,
    "hwVersion": null,
    "protocolType": null,
    "bridgeId": null,
    "status": "OFFLINE",
    "statusDetail": "NOT_ACTIVE",
    "mute": null,
    "supportedSecurity": null,
    "isSecurity": null,
    "signalStrength": null,
    "sigVersion": null,
    "serialNumber": null,
    "batteryLevel": null
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.2 Pushing Device Binding Notifications

Typical Scenario

After an NA subscribes to device binding notifications (the notification type is **bindDevice**) on the IoT platform, the IoT platform sends a notification message to the NA when a device is connected to the IoT platform and bound to the NA.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device binding notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST

URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **bindDevice**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
resultCode	Mandatory	String	body	Indicates the binding result. The value options are **expired** and **succeeded**.
deviceInfo	Optional	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.

Parameter	Mandatory or Optional	Type	Location	Description
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol used by the device.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.

Parameter	Mandatory or Optional	Type	Location	Description
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.
statusDetail	Optional	String(256)	body	Indicates the device status details, which vary according to the value of **status**. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Mandatory or Optional	Type	Location	Description
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "bindDevice",
  "deviceId": "*****",
  "resultCode": "succeeded",
  "deviceInfo": {
    "name": "Sensor_12",
    "manufacturer": "wulian",
    "deviceType": 90,
    "model": "90",
    "mac": "****************",
```

```
"swVersion": "...",
"fwVersion": "...",
"hwVersion": "...",
    "protocolType": "zigbee",
    "description": "smockdetector",
    "nodeType": "GATEWAY"
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.3 Pushing Device Information Change Notifications

Typical Scenario

After an NA subscribes to device information change notifications (the notification type is **deviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device configuration or status (such as manufacturer, location, version and online status) changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device information change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceInfoChanged**.

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chipset provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details about the unique identifiers of chipsets provided by other manufacturers, contact the manufacturers.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol used by the device.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INACTIVE**, and **ABNORMAL**. ● Before the device is connected to the IoT platform for the first time, the device is in the **INACTIVE** state. ● If the device does not report data or send messages to the IoT platform within 25 hours (default value), the device status is **ABNORMAL** (default value). If the device does not report data or send messages to the IoT platform within 49 hours, the device is in the **OFFLINE** state.

Parameter	Mandatory or Optional	Type	Location	Description
statusDetail	Optional	String(256)	Body	Indicates the device status details, which vary according to the value of status. ● When **status** is **ONLINE**, the value can be **NONE**, **CONFIGURATION_PENDING**, **UE_REACHABILITY**, or **AVAILABILITY_AFTER_DDN_FAILURE**. ● When **status** is **OFFLINE**, the value can be **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, **NOT_ACTIVE**, **LOSS_OF_CONNECTIVITY**, or **TIME_OUT**. ● When **status** is **INACTIVE**, the value can be **NONE** or **NOT_ACTIVE**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

NOTE

When the device status information is reported over the southbound API, **status** and **statusDetail** must be included at the same time. In addition, it is recommended that **statusDetail** not be used for logical determination.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType ": "deviceInfoChanged",
  "deviceId": "******************************",
  "gatewayId": "*************",
  "deviceInfo": {
    "name": "Sensor_12",
    "manufacturer": "wulian",
    "type": 90,
    "model": "90",
    "mac": "****************",
"swVersion": "...",
"fwVersion": "...",
"hwVersion": "...",
    "protocolType": "zigbee",
    "description": "smock detector"
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```


1.6.7.4 Pushing Device Data Change Notifications

Typical Scenario

After an NA subscribes to device data change notifications (the notification type is **deviceDataChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of a single service attribute.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device data change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDataChanged**.
requestId	Optional	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
service	Mandatory	DeviceServiceData	body	Indicates service data of the device. For details, see **DeviceServiceData structure**.

DeviceServiceData structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.
eventTime	Mandatory	String	body	Indicates the time when an event occurs. The value is in the format of yyyymmddThhmmssZ. An example value is **20151212T121212Z**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceDataChanged",
  "requestId": "*************",
  "deviceId": "********************************",
```



```
  "gatewayId": "*************",
  "service": {
    "serviceId": "Brightness",
    "serviceType": "Brightness",
    "data": {
      "brightness": 80
    },
    "eventTime": "20170311T163657Z"
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.5 Pushing Batch Device Data Change Notifications

Typical Scenario

After an NA subscribes to batch device data change notifications (the notification type is **deviceDatasChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of multiple service attributes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to batch device data change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDatasChanged**.

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
services	Mandatory	List<DeviceServiceData>	body	Indicates the service list. For details, see **DeviceServiceData structure**.

DeviceServiceData structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.
eventTime	Mandatory	String	body	Indicates the time when an event is reported. The value is in the format of yyyymmddThhmmssZ. An example value is **20151212T121212Z**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceDatasChanged",
  "requestId": "**************",
  "deviceId": "********************************",
  "gatewayId": "**************",
  "service": [
    {
      "serviceId": "Brightness",
      "serviceType": "Brightness",
      "data": {
        "brightness": 80
      },
      "eventTime": "20170311T163657Z"
    },
    {
      "serviceId": "Color",
      "serviceType": "Color",
      "data": {
        "value": "red"
      },
      "eventTime": "20170311T163657Z"
    }
  ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.6 Pushing Device Service Capability Change Notifications

Typical Scenario

After an NA subscribes to device service capability change notifications (the notification type is **deviceCapabilitiesChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the services and properties in the profile file are changed on the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service capability change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST

URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceCapabilitiesChanged**.
appId	Mandatory	String	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the name of the manufacturer of the device model.
manufacturerId	Mandatory	String	body	Identifies the manufacturer of the device model.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol used by the device.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
```



```
  "notifyType": "deviceCapabilitiesChanged",
  "appId": "*****",
  "deviceType": "*************",
  "manufacturerName": "wulian",
  " manufacturerId ": "*************",
  "model": "*************",
  "protocolType": "zigbee"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.7 Pushing Device Service Capability Addition Notifications

Typical Scenario

After an NA subscribes to device service capability addition notifications (the notification type is **deviceCapabilitiesAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is added on the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service capability addition notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceCapabilitiesAdded**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the name of the manufacturer of the device model.
manufacturerId	Mandatory	String	body	Identifies the manufacturer of the device model.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol used by the device.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceCapabilitiesAdded",
  "appId": "*****",
  "deviceType": "*************",
  "manufacturerName": "wulian",
  " manufacturerId ": "*************",
  "model": "*************",
  "protocolType": "zigbee"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


1.6.7.8 Pushing Device Service Capability Deletion Notifications

Typical Scenario

After an NA subscribes to device service capability deletion notifications (the notification type is **deviceCapabilitiesDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service capability deletion notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceCapabilitiesDeleted**.
appId	Mandatory	String	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the name of the manufacturer of the device model.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Mandatory	String	body	Identifies the manufacturer of the device model.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol used by the device.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceCapabilitiesDeleted",
  "appId": "*****",
  "deviceTyp ": "*************",
  " manufacturerName": "*************",
  "manufacturerId ": "*************",
  "model": "*************",
  "protocolType": "*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.9 Pushing Device Deletion Notifications

Typical Scenario

After an NA subscribes to device deletion notifications (the notification type is **deviceDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when the device is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device deletion notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDeleted**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceDeleted",
  "deviceId": "********************************",
  "gatewayId": "*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


1.6.7.10 Pushing Device Acknowledgment Notifications

Typical Scenario

After an NA subscribes to device acknowledgment notifications (the notification type is **messageConfirm**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command acknowledgment message (for example, the command is delivered or executed).

This API applies to devices that use MQTT, for example, devices that have integrated the AgentLite SDK.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device acknowledgment notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **messageConfirm**.
header	Mandatory	Header	body	For details, see the description of the **Header field**.
body	Mandatory	ObjectNode	body	Based on the service definition, an acknowledgment message can carry information such as status change.

Description of the **header** field:

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: /devices/{deviceId} ● Request initiated by a device service: /devices/{deviceId}/services/{serviceId}
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
status	Mandatory	String(1-32)	body	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been received. ● **executed**: The command has been executed.
timestamp	Mandatory	String(1-32)	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "messageConfirm",
  "header": {
    "requestId": "**********",
    "from": "**********",
    "to": "**********",
    "status": "delivered",
    "timestamp": "20151212T121212Z"
  },
  "body": {
```



```
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.11 Pushing Device Command Response Notifications

Typical Scenario

After an NA subscribes to device command response notifications (the notification type is **commandRsp**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command response message (for example, the command execution succeeds or fails).

This API applies to devices that use MQTT, for example, devices that have integrated the AgentLite SDK.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device command response notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **commandRsp**.
header	Mandatory	Header	body	For details, see the description of the **Header field**.

Parameter	Mandatory or Optional	Type	Location	Description
body	Mandatory	ObjectNode	body	Indicates the content of a response message.

Header field description

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: /devices/{deviceId} ● Request initiated by a device service: /devices/{deviceId}/services/{serviceId}
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
serviceType	Mandatory	String	body	Indicates the type of the service to which a command belongs.
method	Mandatory	String(1-128)	body	Indicates a stored response command, for example, **INVITE-RSP**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
```



```
Content-Type: application/json
Body:
{
  "notifyType": "commandRsp",
  "header": {
    "requestId": "***********",
    "from": "***********",
    "to": "***********",
    "deviceId": "***********",
    "serviceType": "Camera",
    "method": "MUTE_COMMANDS"
  },
  "body": {

  }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.12 Pushing Command Status Change Notifications

Typical Scenario

When an NA creates a device command with the callback URL specified, the IoT platform pushes a notification message to the NA if the command status changes (failed, successful, timeout, sent, or delivered).

API Function

The IoT platform pushes notification messages to NAs when the command status changes.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the command request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
commandId	Mandatory	string	body	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
result	Mandatory	CommandResultForDevice	body	Indicates the command status information. For details, see the **CommandResultForDevice structure**.

CommandResultForDevice structure

Parameter	Mandatory or Optional	Type	Location	Description
resultCode	Mandatory	String	body	Indicates the command status.
resultDetail	Mandatory	ObjectNode	body	Indicates the detailed information about the command output.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
request:
Header:
Content-Type: application/json
Body:
{
  "deviceId": "92d3f8da-200a-4143-8d0d-591a7e11de6c",
  "commandId": "108a9c71462a48e09426e06e844d47ba",
  "result": {
    "resultCode": "SENT",
    "resultDetail": null
  }
}
```


Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.13 Pushing Rule Event Notifications

Typical Scenario

After an NA subscribes to rule event notifications (the notification type is **ruleEvent**) on the IoT platform, the IoT platform sends a notification message to the NA when a rule configured on the IoT platform is triggered.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to rule event notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **ruleEvent**.
author	Mandatory	String	body	Identifies the user that creates the rule. The value of this parameter can contain a maximum of 256 characters.
ruleId	Mandatory	String	body	Identifies a rule instance.
ruleName	Mandatory	String	body	Indicates the name of the rule instance.

Parameter	Mandatory or Optional	Type	Location	Description
logic	Optional	String	body	Indicates the logical relationships between conditions.
reasons	Mandatory	List<Reason>	body	Indicates the trigger reasons, which are corresponding to **conditions**. For details, see **Reason structure**.
triggerTime	Mandatory	String	body	Indicates the time when a rule is triggered.
actionsResults	Mandatory	List<ActionResult>	body	Indicates the action execution results.

Reason structure

Parameter	Mandatory or Optional	Type	Location	Description
satisfactionTime	Mandatory	String	body	Indicates the condition satisfaction time.
type	Mandatory	String	body	Indicates the condition type.
id	Optional	String	body	Identifies a condition.
info	Optional	Json	body	Indicates the information carried by different condition types.
transInfo	Optional	Json	body	Indicates the information filled during event push, which is corresponding to the **transInfo** in the condition of the rule.

ActionResult structure:

Parameter	Mandatory or Optional	Type	Location	Description
type	Mandatory	String	body	Indicates the action type.
id	Optional	String	body	Identifies an action.

Parameter	Mandatory or Optional	Type	Location	Description
exception	Either this parameter or **result** is selected.	String	body	Indicates the exception information that is carried when an exception occurs when the rule engine constructs a request corresponding to the action.
result	Either this parameter or **exception** is selected.	Json	body	Indicates the execution result for an action of the **DEVICE_CMD**, **SMS**, or **EMAIL** type. The format is **statusCode**+**body**.
info	Optional	Json	body	Indicates the information carried by different action types.
transInfo	Optional	Json	body	Indicates the information filled during event push, which is corresponding to the **transInfo** in the action.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "ruleEvent",
  "author": "*****",
  "ruleId": "*************",
  "ruleName": "name",
  "reasons": [
    {
      "satisfactionTime": "yyyyMMddTHHmmssZ",
      "type": "******"
    }
  ],
  "triggerTime": "yyyyMMddTHHmmssZ",
  "actionsResults": [
    {
      "type": "******",
      "exception": "******"
    }
  ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```


1.6.7.14 Pushing Device Shadow Status Change Notifications

Typical Scenario

After an NA subscribes to device shadow status change notifications (the notification type is **deviceDesiredPropertiesModifyStatusChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device shadow on the IoT platform succeeds or fails to synchronize data to the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device shadow status change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDesiredPropertiesModifyStatusChanged**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
serviceId	Mandatory	String	body	Identifies a service.
properties	Mandatory	ObjectNode	body	Indicates data attributes of a device shadow.
status	Mandatory	String	body	Indicates the status. The value options are **DELIVERED** and **FAILED**.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "deviceDesiredPropertiesModifyStatusChanged",
  "deviceId": "******",
  "serviceId": "Device",
  "properties": {
    "Model Number": 1,
    "Serial Number": 2,
    "Firmware Version": "v1.1.0"
  },
  "status": "DELIVERED"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.15 Pushing Software Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to software upgrade status change notifications (the notification type is **swUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the software upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade status change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeStateChangeNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade sub-task.
swUpgradeState	Mandatory	String	body	Indicates the software upgrade status. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "swUpgradeStateChangeNotify",
  "deviceId": "******",
  "appId": "**********",
  "operationId": "**********",
  "subOperationId": "**********",
```



```
  "swUpgradeState": "downloading"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.16 Pushing Software Upgrade Result Notifications

Typical Scenario

After an NA subscribes to software upgrade result change notifications (the notification type is **swUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a software upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade result notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeResultNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Mandatory	String	body	Identifies the application to which the device belongs.

Parameter	Mandatory or Optional	Type	Location	Description
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current software version of the device.
targetVersion	Mandatory	String	body	Indicates the target software version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source software version of the device.
swUpgradeResult	Mandatory	String	body	Indicates the software upgrade result. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the upgrade time.
resultDesc	Mandatory	String	body	Indicates the upgrade result description.
errorCode	Mandatory	String	body	Indicates a status error code reported by the device.
description	Mandatory	String	body	Indicates the description of the cause of error.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "swUpgradeResultNotify",
  "deviceId": "******",
  "appId": "***********",
  "operationId": "***********",
  "subOperationId": "***********",
  "curVersion": "1.3",
  "targetVersion": "1.5",
```



```
  "sourceVersion": "1.0",
  "swUpgradeResult": "SUCCESS",
  "upgradeTime": "***",
  "resultDesc": "***",
  "errorCode": "***",
  "description": "***"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.17 Pushing Firmware Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to firmware upgrade status change notifications (the notification type is **fwUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the firmware upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade status change notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeStateChangeNotify**.

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade sub-task.
step	Mandatory	String	body	Indicates the firmware upgrade status. The value options are **0**, **1**, **2**, and **3**.
stepDesc	Mandatory	String	body	Indicates the upgrade status description. ● **downloading** (corresponding to value **1** of **step**): The device is downloading the firmware package. ● **downloaded** (corresponding to value **2** of **step**): The device has finished downloading the firmware package. ● **updating** (corresponding to value **3** of **step**): The device is being upgraded. ● **idle** (corresponding to value **0** of **step**): The device is in the idle state.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "fwUpgradeStateChangeNotify",
  "deviceId": "******",
  "appId": "***********",
```



```
  "operationId": "**************",
  "subOperationId": "************",
  "step": "1",
  "stepDesc": "downloading"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

1.6.7.18 Pushing Firmware Upgrade Result Notifications

Typical Scenario

After an NA subscribes to firmware upgrade result change notifications (the notification type is **fwUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a firmware upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade result notifications.

Note

The NA must return a response code after receiving a message pushed by the IoT platform.

API Prototype

Method	POST
URL	The URL is determined by **callbackUrl** in the subscription request sent by the NA.
Transport Protocol	HTTPS/HTTP

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeResultNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
appId	Mandatory	String	body	Identifies the application to which the device belongs.

Parameter	Mandatory or Optional	Type	Location	Description
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current firmware version of the device.
targetVersion	Mandatory	String	body	Indicates the target firmware version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source firmware version of the device.
status	Mandatory	String	body	Indicates the upgrade result. ● SUCCESS ● FAIL
statusDesc	Mandatory	String	body	Indicates the upgrade result description. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the firmware upgrade duration.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type: application/json
Body:
{
  "notifyType": "fwUpgradeResultNotify",
  "deviceId": "******",
  "appId": "***********",
  "operationId": "***********",
  "subOperationId": "***********",
  "curVersion": "1.6",
  "targetVersion": "1.6",
  "sourceVersion": "1.3",
  "status": "SUCCESS",
  "statusDesc": "****",
  "upgradeTime": "****"
}
```


Response Example

```
Response:
Status Code: 200 OK
```

1.7 Command Delivery

1.7.1 Creating Device Commands

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform provides two command delivery modes:

- Immediate delivery: The IoT platform delivers commands to devices immediately after receiving the commands. This ensures real-time performance but does not ensure serialization.
- Cached command delivery: After receiving commands, the IoT platform caches the commands. When the devices are reachable, the IoT platform delivers the commands in sequence. Specifically, the IoT platform delivers the latter command only after receiving the response of the previous command (which is the ACK automatically replied by the module) to ensure serialization instead of real-time performance.

This API applies to devices that use LWM2M/CoAP, for example, NB-IoT devices.

API Function

This API is used by an NA to deliver a command to a device to control the device. Immediate delivery and cached command delivery are supported on the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String(1-64)	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceId	Mandatory	String(64)	body	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	Mandatory	**CommandDTO**	body	Indicates the command information.
callbackUrl	Optional	String(1024)	body	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.

Parameter	Mandatory or Optional	Type	Location	Description
expireTime	Optional	Integer(>=0)	body	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2). If it is set to **0**, the IoT platform will deliver the command to the specific device immediately no matter what command mode is set on the IoT platform (if the device is sleeping or the link has aged, the device cannot receive the command, the IoT platform cannot receive any response from the device, and the command times out in the end). If **expireTime** is not **0**, commands are cached on the IoT platform (in the **PENDING** state) within the period specified by **expireTime**. The commands are delivered only when the device goes online or reports data to the IoT platform.
maxRetransmit	Optional	Integer(0-3)	body	Indicates the maximum number of times the command is retransmitted.

CommandDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as **serviceId** defined in the profile.

Parameter	Mandatory or Optional	Type	Location	Description
method	Mandatory	String(1-128)	body	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Mandatory	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

Response Parameters

Status Code: 201 Created

Parameter	Type	Description
commandId	String(1-64)	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
appId	String(1-64)	Identifies the application to which the device belongs.
deviceId	String(1-64)	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	**CommandDTO**	Indicates the command information.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2).

Parameter	Type	Description
status	String	Indicates the status of the command. ● **PENDING**: The command is cached and has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered to the device.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers a command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command is retransmitted.

Request Example

```
Method: POST
request: https://server:port/iocm/app/cmd/v1.4.0/deviceCommands
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "deviceId": "********",
  "command": {
    "serviceId": "********",
    "method": "********",
    "paras": {
      "paraName1": "paraValue1",
      "paraName2": "paraValue2"
    }
  },
  "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
  "maxRetransmit": ******

}
```


Response Example

```
Response:
Status Code: 201 Created
Content-Type: application/json
Body:
{
  "commandId": "********",
  "appId": "********",
  "deviceId": "********",
  "command": {
    "serviceId": "********",
    "method": "********",
    "paras": {
      "paraName1": "paraValue1",
      "paraName2": "paraValue2"
    }
  },
  "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
  "expireTime": null,
  "status": "PENDDING",

  "creationTime": "20170222T164000Z",
  "executeTime": null,
  "platformIssuedTime": null,
  "deliveredTime": null,
  "issuedTimes": null,
  "maxRetransmit": ******
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100223	Command counts has reached the upLimit.	The number of cached commands reaches the limit. The number of commands in the **PENDING** state does not exceed the limit. The default value is **20**. Recommended handling: If the commands cached on the IoT platform need to be executed, enable the device to report data to trigger delivery of the cache commands. If a command cached on the IoT platform does not need to be executed, call the API used for modifying device commands V4 to change the state of the command from **PENDING** to **CANCELED**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100612	Device is zombie.	The device is a zombie device. (The interval between the current system time and the time when the device went online exceeds the threshold. The default value is seven days.) Recommended handling: Run the command again after the device goes online.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.7.2 Querying Device Commands

Typical Scenario

After an NA delivers a command to a device, the NA can call this API to query the status and content of the delivered command on the IoT platform to check the command execution status.

API Function

This API is used by an NA to query the status and content of delivered commands on the IoT platform. All the commands delivered by the current application in a specified period or all the commands delivered to a specified device can be queried.

API Prototype

Method	GET
URL	https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?pageNo={pageNo}&pageSize={pageSize}&deviceId={deviceId}&startTime={startTime}&endTime={endTime}&appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
pageSize	Optional	Integer[1,1000]	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
deviceId	Optional	String(64)	query	Uniquely identifies a device for which the commands are queried. The value of this parameter is allocated by the IoT platform during device registration.
startTime	Optional	String	query	Indicates the start time. Commands delivered later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Commands delivered earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
pagination	**Pagination**	Indicates the page information.
data	List<**DeviceCommandResp**>	Indicates the device command list.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalSize	long	Indicates the total number of records.

DeviceCommandResp structure

Parameter	Type	Description
commandId	String(1-64)	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
appId	String(1-64)	Identifies the application to which the device belongs.

Parameter	Type	Description
deviceId	String(1-64)	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	**CommandDTO**	Indicates the command information.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **PENDING**: The command is cached and has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered to the device.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers a command.

Parameter	Type	Description
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command is retransmitted.

CommandDTO structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Optional	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?
pageNo=0&pageSize=10&deviceId=******&startTime=20151112T101012Z&endTime=20151212T1
21212Z
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "pagination": {
    "pageNo": 0,
    "pageSize": 20,
    "totalSize": 100
  },
  "data": [
    {
      "commandId": "********",
      "appId": "********",
      "deviceId": "********",
      "command": {
        "serviceId": "********",
```



```
        "method": "********",
        "paras": {
          "paraName1": "paraValue1",
          "paraName2": "paraValue2"
        }
      },
      "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
      "expireTime": null,
      "status": "PENDDING",
      "result": null,
      "creationTime": "20170222T164000Z",
      "executeTime": null,
      "platformIssuedTime": null,
      "deliveredTime": null,
      "issuedTimes": null,
      "maxRetransmit": ******
    },
    {
      "commandId": "********",
      "appId": "********",
      "deviceId": "********",
      "command": {
        "serviceId": "********",
        "method": "********",
        "paras": {
          "paraName1": "paraValue1",
          "paraName2": "paraValue2"
        }
      },
      "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
      "expireTime": null,
      "status": "PENDDING",
      "result": null,
      "creationTime": "20170222T164000Z",
      "executeTime": null,
      "platformIssuedTime": null,
      "deliveredTime": null,
      "issuedTimes": null,
      "maxRetransmit": ******
    }
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: ● Ensure that neither **startTime** nor **endTime** is null and the value of **endTime** is later than that of **startTime**. ● Ensure that **pageNo** is not null and the value of **pageNo** is greater than **0**. ● Ensure that **pageSize** is not null and the value of **pageSize** is greater than **1**.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.7.3 Modifying Device Commands

Typical Scenario

After an NA delivers a command to a device, the IoT platform does not deliver the command to the device for execution if the command is in queue or the device is offline. In this case, the NA can call this API to modify the command status. Currently, the command status can be changed only to **CANCELED**, indicating that the command is canceled.

API Function

This API is used by an NA to modify the status of a specified command. Currently, only the command in the **PENDING** state can be changed and the status can be changed only to **CANCELED**.

API Prototype

Method	PUT
URL	https://server:port/iocm/app/cmd/v1.4.0/deviceCommands/{deviceCommandId}?appId={appId}

Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceCommandId	Mandatory	String	path	Identifies the command whose status is to be modified. The value of this parameter is obtained after the API used for creating device commands is called.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
status	Mandatory	String	body	Indicates the command status. The value **CANCELED** indicates that the command is revoked.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
commandId	String(1-64)	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
appId	String(1-64)	Identifies the application to which the device belongs.
deviceId	String(1-64)	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	**CommandDTO**	Indicates the command information.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **PENDING**: The command is cached and has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered to the device.

Parameter	Type	Description
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers a command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command is retransmitted.

CommandDTO structure

Parameter	Type	Description
serviceId	String(1-64)	Identifies the service corresponding to the command.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

Request Example

```
Method: PUT
Request:
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands/{deviceCommandId}?
appId={appId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "status": "CANCELED"
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "commandId": "********",
  "appId": "********",
  "deviceId": "********",
  "command": {
    "serviceId": "********",
    "method": "********",
    "paras": {
      "paraName1": "paraValue1",
      "paraName2": "paraValue2"
    }
  },
  "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
```



```
  "expireTime": null,
  "status": "PENDDING",
  "result": null,
  "creationTime": "20170222T164000Z",
  "executeTime": null,
  "platformIssuedTime": null,
  "deliveredTime": null,
  "issuedTimes": null,
  "maxRetransmit": ******
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.

HTTP Status Code	Error Code	Error Description	Remarks
200	100434	The device command is not existed.	The device command does not exist. Recommended handling: Check whether the device command ID in the request is correct.
200	100435	The device command already canceled, expired or executed, Cannot cancel.	The device command has been canceled, expired, or executed. It cannot be canceled.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.7.4 Batch Creating Device Commands

If an NA needs to deliver a command to multiple devices, this API can be used to deliver the command to the devices in batches.

This function is implemented by using the **Creating a Batch Task** API to deliver a batch task. When **taskType** is set to **DeviceCmd**, the task is used to deliver a command to devices in batches.

1.7.5 Creating Device Command Revocation Tasks

Typical Scenario

After an NA delivers commands to a device, the IoT platform does not deliver the commands to the device for execution (the commands are in the **PENDING** state) if the commands are in queue or the device is offline. In this case, the NA can call this API to revoke all the undelivered commands of a specified device. Commands that have been delivered cannot be revoked.

API Function

This API is used by an NA to create a command revocation task to revoke all undelivered commands (that is, commands in the **PENDING** state) with the specified device ID on the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
deviceId	Mandatory	String(1-64)	body	Uniquely identifies a device for which a command is to be revoked. The value of this parameter is allocated by the IoT platform during device registration. A command revocation task will revoke all commands delivered to the device.

Response Parameters

Status Code: 201 Created

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Uniquely identifies a device for which a command revocation task is executed. The value of this parameter is allocated by the IoT platform during device registration.

Parameter	Type	Description
status	String	Indicates the revocation task status. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<**DeviceCommandResp**>	Indicates a list of device commands to be revoked.

DeviceCommandResp structure

Parameter	Type	Description
commandId	String(1-64)	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	**CommandDTO**	Indicates the command information.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2).

Parameter	Type	Description
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered to the device.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers a command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command is retransmitted.

CommandDTO structure

Parameter	Type	Description
serviceId	String(1-64)	Identifies the service corresponding to the command.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?appId={appId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "deviceId": "********"
}
```

Response Example

```
Response:
Status Code: 201 Created
Content-Type: application/json
Body:
{
  "taskId": "********",
  "appId": "********",
  "deviceId": "********",
  "status": "WAITTING",
  "totalCount": 1,
  "deviceCommands": [
    {
      "commandId": "********",
      "appId": "********",
      "deviceId": "********",
      "command": {
        "serviceId": "********",
        "method": "********",
        "paras": {
          "paraName1": "paraValue1",
          "paraName2": "paraValue2"
        }
      },
      "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
      "expireTime": null,
      "status": "PENDDING",
      "result": null,
      "creationTime": "20170222T164000Z",
      "executeTime": null,
      "platformIssuedTime": null,
      "deliveredTime": null,
      "issuedTimes": null,
      "maxRetransmit": ******
    }
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.7.6 Querying Command Revocation Tasks

Typical Scenario

After an NA creates a command revocation task, the NA can call this API to query the details and execution status of the task.

API Function

This API is used by an NA to query the information and status of one or more command revocation tasks based on specified conditions on the IoT platform.

API Prototype

Method	GET

URL	https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?pageNo={pageNo}&pageSize={pageSize}&taskId={taskId}&deviceId={deviceId}&status={status}&startTime={startTime}&endTime={endTime}&appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
pageSize	Optional	Integer[1,1000]	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
taskId	Optional	String	query	Identifies a command revocation task.
deviceId	Optional	String	query	Uniquely identifies a device for which a command revocation task is executed. The value of this parameter is allocated by the IoT platform during device registration.
status	Optional	String	query	Indicates the status of the command revocation task.
startTime	Optional	String	query	Indicates the start time. Revocation tasks created later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
endTime	Optional	String	query	Indicates the end time. Revocation tasks created earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
pagination	**Pagination**	Indicates the page information.
data	List<**DeviceCommandCancelTaskResp**>	Indicates the device command list.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalSize	long	Indicates the total number of records, that is, the total number of commands queried in the command revocation task.

DeviceCommandCancelTaskResp structure

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Uniquely identifies a device in command revocation task. The value of this parameter is allocated by the IoT platform during device registration.
status	String	Indicates the revocation task status. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<**DeviceCommandResp**>	Indicates a list of revoked device commands.

DeviceCommandResp structure

Parameter	Type	Description
commandId	String(1-64)	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Uniquely identifies a device to which a command is delivered. The value of this parameter is allocated by the IoT platform during device registration.
command	**CommandDTO**	Indicates the command information.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.

Parameter	Type	Description
expireTime	Integer(>=0)	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered to the device.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers a command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command is retransmitted.

CommandDTO structure

Parameter	Type	Description
serviceId	String(1-64)	Identifies the service corresponding to the command.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.

Parameter	Type	Description
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?
pageNo={pageNo}&pageSize={pageSize}&taskId={taskId}&deviceId={deviceId}&status={st
atus}&startTime={startTime}&endTime={endTime}&appId={appId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "pagination": {
    "pageNo": 0,
    "pageSize": 20,
    "totalSize": 100
  },
  "data": [
    {
      "taskId": "********",
      "appId": "********",
      "deviceId": "********",
      "status": "WAITTING",
      "totalCount": 1,
      "deviceCommands": [
        {
          "commandId": "********",
          "appId": "********",
          "deviceId": "********",
          "command": {
            "serviceId": "********",
            "method": "********",
            "paras": {
              "paraName1": "paraValue1",
              "paraName2": "paraValue2"
            }
          },
          "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
          "expireTime": null,
          "status": "PENDDING",
          "result": null,
          "creationTime": "20170222T164000Z",
          "executeTime": null,
          "platformIssuedTime": null,
          "deliveredTime": null,
          "issuedTimes": null,
          "maxRetransmit": ******
        }
      ]
```



```
    },
    {
      "taskId": "********",
      "appId": "********",
      "deviceId": "********",
      "status": "WAITTING",
      "totalCount": 1,
      "deviceCommands": [
        {
          "commandId": "********",
          "appId": "********",
          "deviceId": "********",
          "command": {
            "serviceId": "********",
            "method": "********",
            "paras": {
              "paraName1": "paraValue1",
              "paraName2": "paraValue2"
            }
          },
          "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
          "expireTime": null,
          "status": "PENDDING",
          "result": null,
          "creationTime": "20170222T164000Z",
          "executeTime": null,
          "platformIssuedTime": null,
          "deliveredTime": null,
          "issuedTimes": null,
          "maxRetransmit": ******
        }
      ]
    }
  ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.7.7 Calling Device Services

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform does not cache commands but delivers commands immediately. If a device is offline, the commands fail to be delivered. The command formats are defined by the NA and the device, and the IoT platform performs encapsulation and transparent transmission for messages over the API.

This API applies to devices that use MQTT, for example, devices that have integrated the AgentLite SDK.

API Function

This API is used by an NA to deliver a command to a device to control the device.

API Prototype

Method	POST
URL	https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/{serviceId}/sendCommand?appId={appId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
deviceId	Mandatory	String(1-64)	path	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
serviceId	Mandatory	String(1-64)	path	Identifies the service corresponding to the command. The value of this parameter must be the same as **serviceId** defined in the profile.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
header	Mandatory	**CommandNA2CloudHeader**	body	Indicates the message header.
body	Optional	ObjectNode	body	Indicates the message body. The content of the JsonObject is a list of key-value pairs. Every key is the **paraName** of a command defined in the profile.

CommandNA2CloudHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Optional	String(0-128)	body	Identifies a command. The value of this parameter must be unique.
mode	Mandatory	Enum	body	Indicates whether an ACK message is required. ● **NOACK**: No ACK message is required. ● **ACK**: An ACK message is required. ● Other values: invalid
from	Optional	String(128)	body	Indicates the address of the message sender. ● Request initiated by an application: /users/{userId} ● Request initiated by an NA: /{serviceName} ● Request initiated by the IoT platform: /cloud/{serviceName}
toType	Optional	Enum	body	Indicates the type of the message recipient. The value options are **CLOUD** and **GATEWAY**. The default value is **GATEWAY**.
to	Optional	String(128)	body	Indicates the address of the message recipient.
method	Mandatory	String(1-32)	body	Indicates the command name. The value of this parameter must be the same as the command name defined in the profile file.
callbackURL	Optional	String(1024)	body	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.

Response Parameters

Status Code: 202 Accepted

Parameter	Type	Description
status	String(128)	Indicates the command status. ● **sent**: The command has been sent. ● delivered: The device has received the command. ● **failed**: The command delivery fails.

Parameter	Type	Description
timestamp	String(128)	Indicates the timestamp when the command is sent. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
requestId	String(128)	Identifies a device command. If **requestId** is carried in a request, the response carries the same **requestId** as the request; if **requestId** is not carried in a request, the IoT platform allocates a sequence number for the response.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/
{serviceId}/sendCommand
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "header": {
    "mode": "ACK",
    "from": "/users/23212121",
    "method": "INVITE-INIT",
    "callbackURL": "http://10.10.10.10:8043/na/iocm/message/confirm"
  },
  "body": {
    "from": "************",
    "sessionID": "**********",
    "sdp": "**********"
  }
}
```

Response Example

```
Response:
Status Code: 202 Accepted
Content-Type: application/json
Body:
{
  "requestId": "************",
  "status": "sent",
  "timestamp": "**********"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the gateway is normal.
200	100432	The device command is muted.	The device command is muted. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	102203	CommandName is invalid.	The command name is invalid. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100450	The gateway is not online.	The gateway is offline. Recommended handling: Check whether the connection between the gateway and the IoT platform is normal.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100444	The serviceType is not exist.	The service type does not exist. Recommended handling: Check whether the service type carried in the API request parameter **toType** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

1.8 Batch Processing

An NA can perform batch operations on devices connected to the IoT platform through the Batch Processing API.

1.8.1 Creating a Batch Task

Typical Scenario

When an NA needs to perform an operation on a batch of devices, the NA can call this API to create a batch task. Currently, the supported batch operations include delivering pending commands to devices in batches.

API Function

This API is used by an NA to create a batch task for devices on the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/batchtask/v1.1.0/tasks
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(64)	body	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
param	Mandatory	**Object Node**	body	Indicates the task parameter, which varies depending on the value of **taskType**.
taskName	Mandatory	String	body	Indicates the task name. The value is a string of a maximum of 256 characters.
taskType	Mandatory	String	body	Indicates the task type. The values include **DeviceCmd**.
timeout	Mandatory	Integer	body	Indicates the timeout duration of a task, in minutes. The value range is 10 - 2880.
tags	Optional	List<**TagDTO2**>	body	Indicates the tag list.

ObjectNode structure

Parameter	Mandatory or Optional	Type	Location	Description
type	mandatory	String	body	Indicates the batch command type. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	Conditionally mandatory	List<String>	body	Indicates the device ID list. This parameter is mandatory when **type** is set to **DeviceList**.
deviceType	Conditionally mandatory	String	body	Indicates the device type. This parameter is mandatory when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Conditionally optional	String	body	Identifies a manufacturer. This parameter is mandatory when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
model	Conditionally optional	String	body	Indicates the model of the device. This parameter is mandatory when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
deviceLocation	Conditionally mandatory	String	body	Indicates the location of the device. This parameter is mandatory when **type** is set to **DeviceArea**.
groupList	Conditionally mandatory	List<String>	body	Indicates the group ID list or device group name list. When **type** is set to **GroupIdList**, set this parameter to the group ID. When **type** is set to **GroupList**, set this parameter to the device group name.
command	mandatory	**CommandDTO**	body	Indicates the command information.
callbackUrl	optional	String	body	Indicates the push address of the command execution result.
maxRetransmit	optional	Integer(0-3)	body	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	optional	String	body	Indicates the group tag.

CommandDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.

Parameter	Mandatory or Optional	Type	Location	Description
method	Mandatory	String(1-128)	body	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Optional	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
taskID	String	Identifies a batch task.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/batchtask/v1.1.0/tasks
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "appId": "******",
  "timeout": 1000,
```



```
  "taskName": "******",
  "taskType": "DeviceCmd",
  "param": {
    "type": "******",
    "deviceList": [
      "********",
      "********",
      "********"
    ],
    "command": {
      "serviceId": "********",
      "method": "********",
      "paras": {
        "paraName1": "paraValue1",
        "paraName2": "paraValue2"
      }
    }

  }
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "taskID": "******"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	101001	Resource doesn't exist.	The resource does not exist.
200	105001	The batchTask count has reached the limit.	If the number of unfinished tasks is greater than or equal to 10, a message is returned, indicating that the number of tasks reaches the limit.
200	105002	The batchTask name has exist.	The task name exists. Recommended handling: Change the task name.

HTTP Status Code	Error Code	Error Description	Remarks
400	105201	The tagName and tagValue has been used on the platform.	The tagName and tagValue have been used on the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100028	The user has no right.	The user has no operation permission.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105202	The tag is not existed.	The tag does not exist.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.8.2 Querying Information About a Specified Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about the batch task, including the task status and the subtask (task performed for a device) completion status.

API Function

This API is used by an NA to query the information about a single batch task by task ID.

API Prototype

Method	GET
URL	https://server:port/iocm/app/batchtask/v1.1.0/tasks/{taskId}?appId={appId}&select={select}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
taskId	Mandatory	String	path	Identifies a batch task. The value of this parameter is obtained after the batch task is created.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
select	Optional	String	query	Indicates an optional return value. The value can be **tags**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
appId	String	Indicates the **appId** to which the batch task belongs.
taskId	String	Identifies a batch task.
taskName	String	Indicates the name of the batch task.
operator	String	Indicates the operator who delivers the batch task.
taskFrom	String	Indicates the source of the batch task. ● **Portal**: The task is created on the SP portal. ● **Northbound**: The task is created by calling a northbound API.
taskType	String	Indicates the type of the batch task. The values include **DeviceCmd**.

Parameter	Type	Description
status	String	Indicates the status of the batch task. The value options are **Pending**, **Running**, **Complete**, and **Timeout**.
startTime	String	Indicates the time when the batch task is created.
timeout	Integer	Indicates the time when the batch task times out, in seconds.
progress	Integer	Indicates the progress of the batch task. The value is a permillage ranging from 0 to 1000 and is rounded down.
totalCnt	Integer	Indicates the total number of tasks.
successCnt	Integer	Indicates the number of tasks that are successfully executed.
failCnt	Integer	Indicates the number of tasks that fail to be executed.
timeoutCnt	Integer	Indicates the number of tasks with execution timeout.
expiredCnt	Integer	Indicates the number of expired tasks that are not executed.
completeCnt	Integer	Indicates the number of completed tasks, including successful, failed, and timed out tasks.
successRate	Integer	Indicates the task success rate. The value is a permillage ranging from 0 to 1000 and is rounded down.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.
tags	List<**TagDTO2**>	Indicates the tag list of the batch task.

ObjectNode structure

Parameter	Type	Description
type	String	Indicates the batch command type. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	List<String>	Indicates the device ID list. The value is that returned when **type** is set to **DeviceList**.
deviceType	String	Indicates the type of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
manufacturerId	String	Identifies a manufacturer. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.

Parameter	Type	Description
model	String	Indicates the model of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
deviceLocation	String	Indicates the location of the device. The value is that returned when **type** is set to **DeviceArea**.
groupList	List<String>	Indicates the group name list. The value is that returned when **type** is set to **GroupList**.
command	**CommandDTO**	Indicates the command information. .
callbackUrl	String	Indicates the push address of the command execution result..
maxRetransmit	Integer(0-3)	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	String	Indicates the group tag.

CommandDTO structure

Parameter	Type	Description
serviceId	String(1-64)	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Type	Description
tagName	String(1-128)	Indicates the tag name.
tagValue	String(1-1024)	Indicates the tag value.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/batchtask/v1.1.0/tasks/******?appId=******
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json;
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "taskId": "********",
  "taskName": "********",
  "appId": "********",
  "operator": "********",
  "taskFrom": "********",
  "taskType": "********",
  "status": "********",
  "startTime": "********",
  "timeout": 1000,
  "progress": 100,
  "totalCnt": 100,
  "successCnt": 70,
  "failCnt": 10,
  "timeoutCnt": 10,
  "expiredCnt": 10,
  "completeCnt": 100,
  "successRate": 70,
  "param": {
    "fileId": "******"
  }
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100019	Illegal request.	Invalid request. Recommended handling: Check whether the mandatory parameters in the request are set.
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

1.8.3 Querying Information About a Subtask of a Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about a subtask of the batch task, including the subtask execution status and subtask content.

API Function

This API is used by an NA to query detailed information about a subtask of the batch task.

API Prototype

Method	GET
URL	https://server:port/iocm/app/batchtask/v1.1.0/taskDetails?appId={appId}&taskId={taskId}&status={status}&deviceId={deviceId}&commandId={commandId}&pageNo={pageNo}&pageSize={pageSize}
Transport Protocol	HTTP/HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
appId	Optional	String	query	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform. Set this parameter to the value of **appId** of the authorized application.
taskId	Mandatory	String	query	Identifies a batch task.
status	Optional	String	query	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
deviceId	Optional	String	query	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration.
commandId	Optional	String	query	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records to be displayed on each page. Pagination is implemented based on the value of this parameter. The value is an integer ranging from 1 to 250. The default value is **25**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
totalCount	Long	Indicates the total number of records.
taskDetails	List<QueryTaskDetailDTOCloud2NA>	Indicates the task details list.

QueryTaskDetailDTOCloud2NA structure

Parameter	Type	Description
status	String	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
output	String	Indicates the output of a batch command delivery task.
error	String	Indicates the cause of error, in the format of {\"error_code\": \"****\", \"error_desc\":\"*****\"}.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.

ObjectNode structure

Parameter	Type	Description
deviceId	String	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration. The value is that returned when **taskType** is set to **DeviceCmd**.
commandId	String	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/batchtask/v1.1.0/taskDetails?
appId=********&taskId=******&status=******&pageNo=******&pageSize=******
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "totalCount": 1,
  "pageNo": 0,
  "pageSize": 1,
  "taskDetails": [
    {
      "status": "WaitResult",
      "output": "{\"requestId\":\"*****\",\"commandResult\":null}",
      "error": null,
      "param": {
        "deviceId": "*******",
        "commandId": "*******"
      }
    },
    {
      "status": "WaitResult",
      "output": "{\"requestId\":\"******\",\"commandResult\":null}",
      "error": null,
      "param": {
        "deviceId": "******",
        "commandId": "*******"
      }
    }
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9 Device Group Management

1.9.1 Creating a Device Group

Typical Scenario

An NA can call this API to create device groups on the IoT platform, and allocate devices to different device groups for group management. A device can be bound to multiple device groups.

When the NA needs to perform operations on devices (such as upgrading device software and firmware or delivering commands to devices in batches), the NA can select devices to be operated by device group.

API Function

This API is used by an NA to create device groups on the IoT platform to manage devices by group.

API Prototype

Method	POST
URL	https://server:port/iocm/app/devgroup/v1.3.0/devGroups
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.
appId	Optional	String (50)	body	This parameter must be specified when you want to add a device group under an authorized application. Set this parameter to the ID of the authorized application.
maxDevNum	Optional	Integer (>=0)	body	Indicates the maximum number of devices in the device group. The default value is **0**. If the value is **0**, the number of devices is not limited.
deviceIds	Optional	List<String>	body	Identifies the devices to be added to the device group.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group.

Parameter	Type	Description
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer (>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.
deviceIds	List<String>	Identifies the devices to be added to the device group.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/devgroup/v1.3.0/devGroups
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "name": "********",
  "description": "******",
  "appId": "********",
  "maxDevNum": "********",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```

Response Example

```
Response:
Status Code: 201 OK
Content-Type: application/json
Body:
{
  "name": "********",
  "description": "******",
  "id": "********",
  "appId": "********",
  "maxDevNum": "********",
  "curDevNum": "********",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100607	The devGroup has reached the limit.	The number of device groups reaches the limit. Recommended handling: Check whether the number of created device groups reaches the upper limit specified in the license.
400	100609	Too much devices to add.	Too many devices are added to the device group. Recommended handling: Ensure that the number of device IDs contained in **deviceIds** is within the range specified by **maxDevNum**.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.2 Deleting a Device Group

Typical Scenario

If a device group is no longer needed on the IoT platform due to group changes, an NA can call this API to delete a specified device group.

API Function

This API is used by an NA to delete a specified device group based on group ID from the IoT platform.

API Prototype

Method	DELETE
URL	https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?accessAppId={accessAppId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	This parameter must be specified when you want to delete a device group under an authorized application. Set this parameter to the ID of the authorized application.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: DELETE
Request:
https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?
accessAppId={accessAppId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.3 Modifying a Device Group

Typical Scenario

If information about a device group (such as the device group name and the device quantity limit in the device group) needs to be modified due to service changes, an NA can call this API to modify the information.

API Function

This API is used by an NA to modify information about a specified device group on the IoT platform.

API Prototype

Method	PUT
URL	https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?accessAppId={accessAppId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
accessAppId	Optional	String	query	This parameter must be specified when you want to modify information about a device group under an authorized application. Set this parameter to the ID of the authorized application.
devGroupId	Mandatory	String (50)	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.

Parameter	Mandatory or Optional	Type	Location	Description
maxDevNum	Optional	Integer(>=0)	body	Indicates the maximum number of devices in the device group. The default value is **0**. If the value is **0**, the number of devices is not limited.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group.
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.

Request Example

```
Method: PUT
Request:
https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?
accessAppId={accessAppId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "name": "********",
  "description": "******",
  "maxDevNum": "********"
}
```

Response Example

```
Response:
Status Code: 200 OK
```



```
Content-Type: application/json
Body:
{
  "name": "********",
  "description": "******",
  "id": "********",
  "appId": "********",
  "maxDevNum": "********",
  "curDevNum": "********"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.4 Querying the Device Group List

Typical Scenario

An NA can call this API to query information of all the created device groups to check the group details and usage of the device groups.

API Function

This API is used by an NA to query information about all created device groups on the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/devgroup/v1.3.0/devGroups?accessAppId={accessAppId}&pageNo={pageNo}&pageSize={pageSize}&name={name}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
accessAppId	Optional	String	query	This parameter must be specified when you want to query information about a device group under an authorized application. Set this parameter to the ID of the authorized application.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of device group records on each page. The default value is **1**.
name	Optional	String	query	Indicates the device group name.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	long	Indicates the total number of device groups.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of device group records on each page.
list	List<**QueryDevGroupDTOCloud2NA**>	Indicates the device group details.

QueryDevGroupDTOCloud2NA structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group.
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.
creator	String(1-50)	Indicates the name of the user who creates the device group.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/devgroup/v1.3.0/devGroups?
accessAppId={accessAppId}&pageNo={pageNo}&pageSize={pageSize}&name={name}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "totalCount": "********",
  "pageNo": "******",
  "pageSize": "********",
  "list": [
    "object"
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.5 Querying Information About a Specified Device Group

Typical Scenario

An NA can call this API to query information about a specified device group to check the usage of the device group.

API Function

This API is used by an NA to query information about a specified device group based on group ID on the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?accessAppId={accessAppId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	This parameter must be specified when you want to query information about a device group under an authorized application. Set this parameter to the ID of the authorized application.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group.
appId	String (50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group.
curDevNum	Integer	Indicates the current number of devices in the device group.
creator	String(1-50)	Indicates the name of the user who creates the device group.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/devgroup/v1.3.0/devGroups/{devGroupId}?
accessAppId={accessAppId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "name": "********",
  "description": "******",
  "id": "********",
  "appId": "******",
  "maxDevNum": "******",
  "curDevNum": "******",
  "creator": "******"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.6 Querying Members in a Specified Device Group

Typical Scenario

An NA can call this API to query information about members in a specified device group.

API Function

This API is used by an NA to query devices in a specified device group based on group ID on the IoT platform.

API Prototype

Method	GET
URL	https://server:port/iocm/app/dm/v1.2.0/devices/ids?devGroupId={devGroupId}&accessAppId={accessAppId}&pageNo={pageNo}&pageSize={pageSize}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
devGroupId	Mandatory	String	query	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
appId	Optional	String	query	This parameter must be specified when you want to query information about a device group under an authorized application. Set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer(1000)	query	Indicates the number of devices on each page. The default value is **10**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
totalCount	long	Indicates the number of devices in the device group.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of devices on each page.
deviceIds	List<String>	Identifies the devices in the device group.

Request Example

```
Method: GET
Request:
https://server:port/iocm/app/dm/v1.2.0/devices/ids?
devGroupId={devGroupId}&accessAppId={accessAppId}&pageNo={pageNo}&pageSize={pageSi
ze}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "totalCount": "********",
  "pageNo": "******",
  "pageSize": "********",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	107001	The serviceId is not exist.	The service ID does not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.7 Adding Members to a Device Group

Typical Scenario

An NA can call this API to add a new device or an existing device to a specified device group. Before adding a device to a device group, you are advised to query the current number of devices and the maximum number of devices allowed in the device group by calling the API for querying information about a specified device group.

API Function

This API is used by an NA to add devices to a specified device group on the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/dm/v1.1.0/devices/addDevGroupTagToDevices?accessAppId={accessAppId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
accessAppId	Optional	String	query	This parameter must be specified when you want to add members to a device group under an authorized application. Set this parameter to the ID of the authorized application.
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies the devices to be added to the device group.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
devGroupId	String(1-50)	Identifies a device group.
deviceIds	List<String>	Identifies the devices to be added to the device group.

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/dm/v1.1.0/devices/addDevGroupTagToDevices?
accessAppId={accessAppId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
```



```
  "devGroupId": "******",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "devGroupId": "******",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.9.8 Deleting Members from a Device Group

Typical Scenario

If one or more devices in a device group do not belong to the device group any longer, an NA can call this API to delete them from the device group.

API Function

This API is used by an NA to delete devices from a specified device group on the IoT platform.

API Prototype

Method	POST
URL	https://server:port/iocm/app/dm/v1.1.0/devices/deleteDevGroupTagFromDevices?accessAppId={accessAppId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
accessAppId	Optional	String	query	This parameter must be specified when you want to delete members from a device group under an authorized application. Set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies devices to be deleted from the device group.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: POST
Request:
https://server:port/iocm/app/dm/v1.1.0/devices/deleteDevGroupTagFromDevices?
accessAppId={accessAppId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "devGroupId": "******",
  "deviceIds": [
    "********",
    "********",
    "********"
  ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.

HTTP Status Code	Error Code	Error Description	Remarks
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

1.10 Device Upgrade

1.10.1 Querying a Version Package List

Typical Scenario

Before upgrading the device version, an NA can call this API to query the version upgrade packages that have been uploaded to the IoT platform to ensure that the target version package has been uploaded.

API Function

This API is used by an NA to query a list of uploaded version packages that meet a specified condition.

API Prototype

Method	GET
URL	https://server:port/iodm/northbound/v1.5.0/category?fileType={fileType}&deviceType={deviceType}&model={model}&manufactureName={manufactureName}&version={version}&pageNo={pageNo}&pageSize={pageSize}

Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
fileType	Optional	String(256)	query	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	Optional	String(256)	query	Indicates the type of the device to which the version package is applicable.
model	Optional	String(256)	query	Indicates the model of the device to which the version package is applicable.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device to which the version package is applicable.
version	Optional	String(256)	query	Indicates the version of the version package.
pageNo	Optional	Integer	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
data	List<CategoryInfo>	Indicates the information about a list of version packages.
pageNo	Integer	Indicates the page number.
pageSize	Integer	Indicates the number of records to be displayed on each page.
totalCount	Integer	Indicates the total number of query results.

CategoryInfo structure

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.
description	String	Indicates the version package description.
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Request Example

```
Method: GET
Request:
```

```
https://server:port/iodm/northbound/v1.5.0/category?
fileType={fileType}&deviceType={deviceType}&model={model}&manufactureName={manufac
tureName}&version={version}&pageNo={pageNo}&pageSize={pageSize}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "data":[
        {
            "fileId":"****************",
            "name":"******",
            "version":"V1.1.10",
            "fileType":"softwarePackage",
            "deviceType":"******",
            "model":"******",
            "manufacturerName":"***",
            "protocolType":"CoAP",
            "description":"Test software package made by WYH",
            "date":"2017-08-11",
            "uploadTime":"20151212T121212Z"
        },
        {
            "fileId":"****************",
            "name":"******",
            "version":"1.0",
            "fileType":"firmwarePackage",
            "deviceType":"WaterMeter",
            "model":"17",
            "manufacturerName":"******",
            "protocolType":"CoAP",
            "description":null,
            "date":" 2017-11-11",
            "uploadTime":"20151212T121212Z"
        }
    ],
    "pageNo":0,
    "pageSize":2,
    "totalCount":2
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the values of **pageNo** and **pageSize** in the API request are within the valid ranges.

HTTP Status Code	Error Code	Error Description	Remarks
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

1.10.2 Querying a Specified Version Package

Typical Scenario

Before upgrading the device version, an NA can call this API to query the target version upgrade package to ensure that it has been uploaded to the IoT platform.

API Function

This API is used by an NA to query a specified version package based on the version package ID on the IoT platform. The version package ID can be obtained by calling the Querying a Version Package List API.

API Prototype

Method	GET
URL	https://server:port/iodm/northbound/v1.5.0/category/{fileId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
fileId	Mandatory	String	path	Identifies a version package. The version package ID can be obtained by calling the API for querying a version package list.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.
description	String	Indicates the version package description.

Parameter	Type	Description
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Request Example

```
Method: GET
Request:
https://server:port/iodm/northbound/v1.5.0/category/{fileId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "fileId": "**********",
  "name": "******",
  "version": "V1.1.10",
  "fileType": "softwarePackage",
  "deviceType": "******",
  "model": "******",
  "manufacturerName": "***",
  "protocolType": "CoAP",
  "description": "Test software package made by WYH",
  "date": "2015-2-2 ",
  "uploadTime": "20151212T121212Z"
}
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

1.10.3 Deleting a Specified Version Package

Typical Scenario

If a device version package is no longer needed, an NA can call this API to delete the version package from the IoT platform.

API Function

This API is used by an NA to delete a specified version package from the IoT platform based on the version package ID. The ID of the version package to be deleted can be obtained by calling the Querying a Version Package List API.

API Prototype

Method	DELETE
URL	https://server:port/iodm/northbound/v1.5.0/category/{fileId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
fileId	Mandatory	String	path	Identifies a version package. The version package ID can be obtained by calling the API for querying a version package list.

Response Parameters

Status Code: 200 OK

Request Example

```
Method: DELETE
Request:
https://server:port/iodm/northbound/v1.5.0/category/{fileId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

1.10.4 Creating a Software Upgrade Task

Typical Scenario

If the device software needs to be upgraded, an NA can call this API to create a software upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the software of multiple devices on the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Prototype

Method	POST
URL	https://server:port/iodm/northbound/v1.5.0/operations/softwareUpgrade
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
fileId	Mandatory	String	body	Identifies the target software version.
targets	Mandatory	**OperateDevices**	body	Indicates the target to be upgraded.
policy	Optional	**OperatePolicy**	body	Indicates the execution policy of the upgrade task.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	Optional	String(256)	body	Indicates the device type. This parameter must be specified when a device group is specified.
model	Optional	String(256)	body	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer[1, 5]	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
operationId	String	Identifies an operation task.

Request Example

```
Method: POST
Request:
https://server:port/iodm/northbound/v1.5.0/operations/softwareUpgrade
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "fileId": "**********",
  "targets": {
    "devices": [
      "*****"
    ]
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "operationId": "**********"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.

HTTP Status Code	Error Code	Error Description	Remarks
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

1.10.5 Creating a Firmware Upgrade Task

Typical Scenario

If the device firmware needs to be upgraded, an NA can call this API to create a firmware upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the firmware of multiple devices on the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Prototype

Method	POST
URL	https://server:port/iodm/northbound/v1.5.0/operations/firmwareUpgrade
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
fileId	Mandatory	String	body	Identifies the target firmware version.
targets	Mandatory	**OperateDevices**	body	Indicates the target to be upgraded.
policy	Optional	**OperatePolicy**	body	Indicates the execution policy of the upgrade task.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	Optional	String(256)	body	Indicates the device type. This parameter must be specified when a device group is specified.
model	Optional	String(256)	body	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer[1,5]	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
operationId	String	Identifies an operation task.

Request Example

```
Method: POST
Request:
https://server:port/iodm/northbound/v1.5.0/operations/firmwareUpgrade
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
Body:
{
  "fileId": "**********",
  "targets": {
    "devices": [
      "*****"
    ]
  }
}
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "operationId": "**********"
}
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.

HTTP Status Code	Error Code	Error Description	Remarks
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.

HTTP Status Code	Error Code	Error Description	Remarks
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

1.10.6 Querying Details About a Specified Task

Typical Scenario

After a device software or firmware upgrade task is created, an NA can call this API to query details about the upgrade task, including the configuration and execution status.

API Function

This API is used by an NA to query details about a software or firmware upgrade task, including the configuration and execution status.

API Prototype

Method	GET

URL	https://server:port/iodm/northbound/v1.5.0/operations/{operationId}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
operationId	String	Identifies an operation task.
createTime	String	Indicates the time when the operation task is created. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
startTime	String	Indicates the time when the operation task is started. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Type	Description
stopTime	String	Indicates the time when the operation task is stopped. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
targets	**OperateDevices**	Indicates the target device on which the operation is performed.
policy	**OperatePolicy**	Indicates the operation execution policy.
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	**OperationStaResult**	Indicates the statistics on operation results.
extendPara	Map<String, String>	Indicates extended information of the operation task. The value of this parameter varies depending on the operation type.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Indicates the device type. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Type	Description
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The operation task is executed now. ● **device_online**: The operation task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.

Parameter	Type	Description
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

Request Example

```
Method: GET
Request:
https://server:port/iodm/northbound/v1.5.0/operations/{operationId}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "operationId": "**********",
  "createTime": "20151212T121212Z",
  "startTime": "20151212T121212Z",
  "stopTime": null,
  "operateType": "software_upgrade",
  "targets": {
    "deviceGroups": null,
    "deviceType": "******",
    "model": "******",
    "manufacturerName": "***",
    "devices": [
      "**********"
    ]
  },
  "policy": null,
  "status": "FAIL",
  "staResult": {
    "wait": 0,
    "processing": 0,
    "success": 0,
    "fail": 1,
    "stop": 0,
    "timeout": 0
  },
  "extendPara": {
```



```
    "fileVersion": "V1.1.10"
  }
}
```

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

1.10.7 Querying Details About Subtasks of a Specified Task

Typical Scenario

After a device software or firmware upgrade task is created, the upgrade of each device involved in the task is a subtask (the number of subtasks is the same as that of the devices involved in the task). An NA can call this API to query details about subtasks of the upgrade task to check their execution status.

API Function

This API is used by an NA to query upgrade status of each device involved in a software or firmware upgrade task.

API Prototype

Method	GET
URL	https://server:port/iodm/northbound/v1.5.0/operations/{operationId}/subOperations?subOperationStatus={subOperationStatus}&pageNo={pageNo}&pageSize={pageSize}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
subOperationStatus	Optional	String	query	Indicates subtask status. If this parameter is not specified, execution details of all subtasks under the operation task are queried. ● **wait**: The subtask is waiting to be executed. ● **processing**: The subtask is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The subtask is successfully executed. ● **stop**: The subtask stops being executed.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer(>=0)	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer[1, 100]	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
data	List<**SubOperationInfo**>	Indicates information about a subtask list.
pageNo	Integer(>=0)	Indicates the page number.
pageSize	Integer[1,100]	Indicates the number of records to be displayed on each page.
totalCount	Integer(>=0)	Indicates the total number of query results.

SubOperationInfo structure

Parameter	Type	Description
subOperationId	String	Identifies a subtask.
createTime	String	Indicates the time when the subtask is created. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
startTime	String	Indicates the time when the subtask is started. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
stopTime	String	Indicates the time when the subtask is stopped. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Type	Description
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
deviceId	String	Uniquely identifies a device for which the subtasks are queried. The value of this parameter is allocated by the IoT platform during device registration.
status	String	Indicates the subtask status. ● **wait**: The subtask is waiting to be executed. ● **processing**: The subtask is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The subtask is successfully executed. ● **stop**: The subtask stops being executed.
detailInfo	String	Indicates detailed description of the subtask status. In case of a failure, the value of this parameter is the failure cause.
extendPara	Map<String, String>	Indicates extended information of the subtask. The value of this parameter varies depending on the operation type.

Request Example

```
Method: GET
Request:
https://server:port/iodm/northbound/v1.5.0/operations/{operationId}/subOperations?
subOperationStatus={subOperationStatus}&pageNo={pageNo}&pageSize={pageSize}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```

Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "totalCount": 1,
  "pageNo": 0,
  "pageSize": 1,
  "date": [
    {
      "subOperationId": "**********",
      "createTime": "20151212T121212Z",
      "startTime": "20151212T121212Z",
      "stopTime": null,
      "operateType": "software_upgrade",
      "deviceId": "**********",
      "status": "FAIL",
      "detailInfo": "The task failed to start, unable to find protocol service
based on device information",
```

```
      "extendInfo": {
        "upgradeTime": null,
        "sourceVersion": null,
        "curVersion": "V1.1.10",
        "downloadTime": null,
        "targetVersion": null
      }
    }
  ]
}
```

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

1.10.8 Queries a Task List

Typical Scenario

An NA can call this API to query the created upgrade tasks to view the detailed information and execution status of each upgrade task.

API Function

This API is used by an NA to query details about upgrade tasks that meet specified conditions.

API Prototype

Method	GET
URL	https://server:port/iodm/northbound/v1.5.0/operations?operationType={operationType}&operationStatus={operationStatus}&deviceType={deviceType}&manufacturerName={manufacturerName}&model={model}&deviceId={deviceId}&pageNo={pageNo}&pageSize={pageSize}
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory or Optional	Type	Location	Description
app_key	Mandatory	String	header	Identifies an application that can be accessed on the IoT platform. The value of this parameter is allocated by the IoT platform when the application is created on the platform.
Authorization	Mandatory	String	header	Indicates the authentication information for accessing the IoT platform. The value is **Bearer {accessToken}**, in which **{accessToken}** indicates the access token returned by the Authentication API.
operationType	Optional	String(256)	query	Indicates the operation type. ● firmware_upgrade ● software_upgrade

Parameter	Mandatory or Optional	Type	Location	Description
operationStatus	Optional	String(256)	query	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
deviceType	Optional	String(256)	query	Indicates the device type for which the operation task is intended.
model	Optional	String(256)	query	Indicates the device model for which the operation task is intended.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device for which the operation task is intended.
deviceId	Optional	String(256)	query	Uniquely identifies a device for which the operation task is intended. The value of this parameter is allocated by the IoT platform during device registration.
pageNo	Optional	Integer(>=0)	query	Indicates the page number to be queried. The value is an integer greater than or equal to 0. The default value is **0**, indicating that the first page is queried.
pageSize	Optional	Integer[1, 100]	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

Status Code: 200 OK

Parameter	Type	Description
data	List<**OperationInfo**>	Indicates information about a task list.
pageNo	Integer(>=0)	Indicates the page number.

Parameter	Type	Description
pageSize	Integer[1,100]	Indicates the number of records to be displayed on each page.
totalCount	Integer(>=0)	Indicates the total number of query results.

OperationInfo structure

Parameter	Type	Description
operationId	String	Identifies an operation task.
createTime	String	Indicates the time when the operation task is created. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
startTime	String	Indicates the time when the operation task is started. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
stopTime	String	Indicates the time when the operation task is stopped. The time format is yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
targets	**OperateDevices**	Indicates the target device on which the operation is performed.
policy	**OperatePolicy**	Indicates the operation execution policy.
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	**OperationStaResult**	Indicates the statistics on operation results.

Parameter	Type	Description
extendPara	Map<String, String>	Indicates extended information of the operation task. The value of this parameter varies depending on the operation type.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Indicates the device type. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The operation task is executed now. ● **device_online**: The operation task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Type	Description
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

Request Example

```
Method: GET
Request:
https://server:port/iodm/northbound/v1.5.0/operations?
operationType={operationType}&operationStatus={operationStatus}&deviceType={device
Type}&manufacturerName={manufacturerName}&model={model}&deviceId={deviceId}&pageNo
={pageNo}&pageSize={pageSize}
Header:
app_key: ******
Authorization: Bearer ******
Content-Type: application/json
```


Response Example

```
Response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "totalCount": 1,
  "pageNo": 0,
  "pageSize": 1,
  "data": [
    {
      "operationId": "**********",
      "createTime": "20151212T121212Z",
      "startTime": "20151212T121212Z",
      "stopTime": null,
      "operateType": "software_upgrade",
      "targets": {
        "deviceGroups": null,
        "deviceType": "******",
        "model": "******",
        "manufacturerName": "***",
        "devices": [
          "**********"
        ]
      },
      "policy": null,
      "status": "FAIL",
      "staResult": {
        "wait": 0,
        "processing": 0,
        "success": 0,
        "fail": 1,
        "stop": 0,
        "timeout": 0
      },
      "extendPara": {
        "fileVersion": "V1.1.10"
      }
    }
  ]
}
```

Error Code

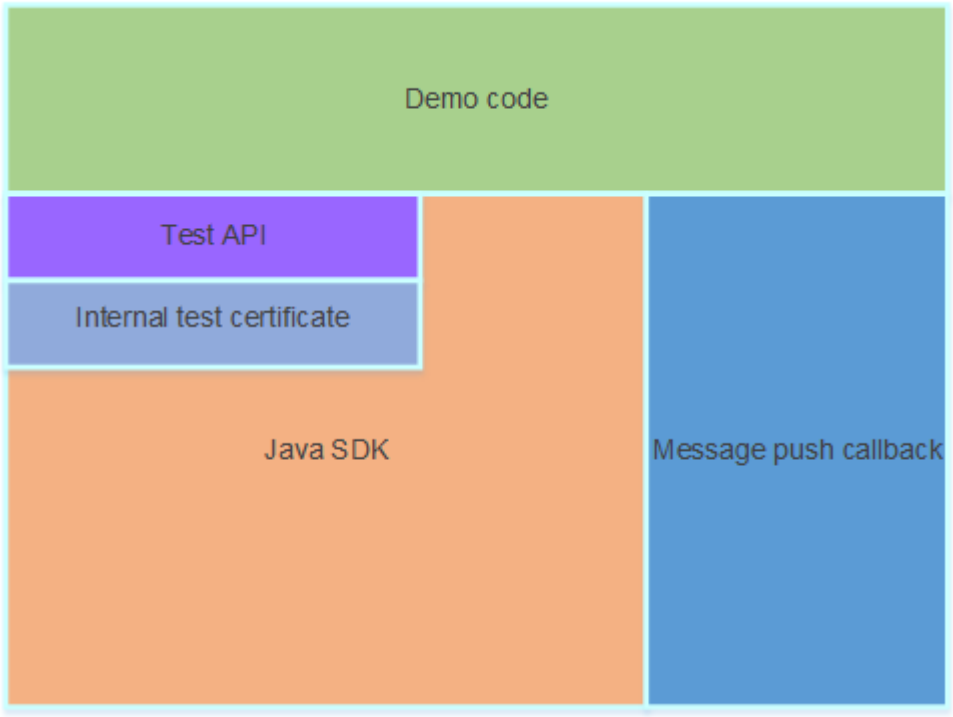
HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

2 Northbound Java SDK API Reference

2.1 SDK Demo Architecture and Usage Guide

- The demo code is the sample code used for calling SDK APIs, including initializing and calling each API. The demo code is for your reference only.
- The Java SDK uses Java methods to call northbound RESTful APIs provided by the IoT platform to communicate with the platform.
- The message push callback uses Java code to implement the callback API, which is called by the IoT platform to push messages to application servers. An application inherits the PushMessageReceiver class and rewrites the methods in the class to receive the content of a push message.
- The test API tests whether the basic functions between the SDK and IoT platform are available and generates test results. During the test, the internal test certificate and the certificate set by developers are used.



2.2 SDK Initialization Configuration and Test

2.2.1 Methods of the NorthApiClient Class

The methods of the NorthApiClient class are used to create an application instance. They are the prerequisites for calling other SDK APIs. The following describes the main methods:

Method	Description
public void setClientInfo(ClientInfo clientInfo)	Initializes the parameters of NorthApiClient. For details on the definition of ClientInfo, see **Methods of the Clientinfo Class**.
public void setHttpConnection(int httpSocketTimeout, int httpConnectionTimeout, int maxConnectionAmount, int maxConnectionAmountPer-Route)	You can set the socket timeout interval (in milliseconds), connection timeout interval (in milliseconds), maximum number of connections, and maximum number of connections for each route in the HTTP connection pool. The default values of the four parameters are **30000**, **30000**, **200**, and **200** respectively. **This method must be called before the initSSLConfig() or initSSLConfig(SSLConfig sslConfig) method is called.**

Method	Description
public void initSSLConfig()	Initializes the two-way authentication configuration. Other methods can only be used after this method is called. **NOTICE** This method uses a test certificate, which is an informal certificate, and does not verify the host name. Therefore, this method is used only in the integration interconnection commissioning phase.
public void initSSLConfig(SSLConfig sslConfig)	Initializes the two-way authentication configuration. Other methods can only be used after this method is called. For details on the definition of SSLConfig, see **SSLConfig Class Method Description**. **NOTICE** This method is used to import certificates and can be used in commercial use. Before calling this method, you can call the setHostnameVerifier(HostnameVerifier hostnameVerifier) method to verify the host name. Otherwise, the strict host name verification is used by default.
public void setHostnameVerifier(HostnameVerifier hostnameVerifier)	Customizes the host name verification method. **This method needs to be called before the initSSLConfig(SSLConfig sslConfig) method is called.** Otherwise, the strict host name verification is used by default. **NOTICE** ● If the host name in the commercial certificate is the same as that in the platform environment, you do not need to call this method. By default, the strict host name verification is used. ● The host name verification method should be based on the security principle. The value **true** should not be returned directly.
public String getVersion()	Queries the SDK version.

2.2.2 Methods of the ClientInfo Class

The methods of the ClientInfo class are used to set basic information for interworking with the IoT platform. The following describes the main methods (excluding the get method):

Method	Description
public void setPlatformIp(String platformIp)	Sets the IP address of the IoT platform.
public void setPlatformPort(String platformPort)	Sets the port number of the IoT platform, for example, 8743.

Method	Description
public void setAppId(String appId)	Sets the application ID. The application ID is an identity allocated by the IoT platform to a third-party application, and is used to uniquely identify an application.
public void setSecret(String secret)	Sets the secret of the application. The secret indicates the password for the application ID on the IoT platform.

2.2.3 Methods of the NorthApiException Class

When the SDK processing is abnormal or the IoT platform fails to process an SDK request, a method of the NorthApiException class is thrown. The following describes the main methods (excluding the set method):

Method	Description
public String geterror_code()	Obtains an error code. If the SDK processing is abnormal, the SDK generates a four-digit error code. If the IoT platform fails to process an SDK request, the IoT platform generates a six-digit error code.
public String getError_desc()	Obtains the description corresponding to the error code.
public String getHttpStatusCode()	Obtains the abnormal HTTP status code. This SDK encapsulates RESTful API using HTTP. Therefore, if an exception occurs when an RESTful API is called by the SDK, an abnormal HTTP status code is returned, such as 404 or 200.
public String getHttpReasonPhase()	Obtains the cause description (ReasonPhrase) corresponding to the abnormal HTTP status code (ReasonPhrase). For example, if the HTTP status code is 400, the cause is "Not Found." If the HTTP status code is 200, the cause is "OK."
public String getHttpMessage()	Obtains the exception content if the IoT platform fails to process a request and the exception does not contain an error code.

2.2.4 Methods of the SSLConfig Class

The methods of the SSLConfig class are used to set the certificate path and password. The following describes the main methods (excluding the get method):

Method	Description
public String setSelfCertPath(String selfCertPath)	Sets the absolute path of a client certificate.
public String setSelfCertPwd(String selfCertPwd)	Sets the password of the client certificate.
public String setTrustCAPath(String trustCAPath)	Sets the absolute path of the server CA certificate.
public String setTrustCAPwd(String trustCAPwd)	Sets the password of the server CA certificate.

2.2.5 Methods of the BasicTest Class

The methods of the BasicTest class are used to test whether basic functions, such as certificate, login, and device registration, are normal. The method is as follows:

Method	Description
public static void beginBasicTest(String platformIp, String platformPort, String appId, String secret, SSLConfig sslConfig)	After setting the platform IP address, port number, application ID, secret, and commercial certificate (optional), you can use this method to test whether basic functions, such as certificate, login, and device registration, are normal. If any exception occurs, the cause and suggestion are displayed on the console.

2.3 Service API List

2.3.1 Secure Application Access

After an NA obtains authentication information and connects to the IoT platform, the NA uses the authentication information to call other APIs.

2.3.1.1 Authentication

Typical Scenario

This API is called by an NA for access authentication when the NA accesses open APIs of the IoT platform for the first time. After the authentication of the NA expires, the NA must call this API to perform authentication again so that the NA can continue to access open APIs of the IoT platform.

API Function

This API is used by an NA to get authenticated before accessing open APIs of the IoT platform for the first time.

Note

The authentication API is the prerequisite for calling other APIs. Except the authentication API itself, other APIs must use the accessToken obtained by the authentication API.

If the accessToken is obtained for multiple times, the previous accessToken is invalid and the last accessToken obtained is valid. Do not obtain the accessToken through concurrent attempts.

API Description

```
AuthOutDTO getAuthToken() throws NorthApiException
```

Class

Authentication

Parameter Description

The application ID and secret use the values of **Methods of the ClientInfo Class** in the **NorthApiClient class**.

Return Value

AuthOutDTO

Parameter	Type	Description
scope	String(256)	Indicates the application permission scope, that is, the scope of IoT platform resources that can be accessed using the **accessToken**. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the type of the **accessToken**. This parameter has a fixed value of **bearer**.
expiresIn	Integer(256)	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.1.2 Refreshing a Token

Typical Scenario

The access token obtained by calling the Authentication API has a valid time. When the access token is about to expire, an NA can call this API to obtain a new access token.

API Function

This API is used by an NA to obtain a new access token from the IoT platform when the access token is about to expire.

API Description

```
AuthRefreshOutDTO refreshAuthToken(AuthRefreshInDTO arInDTO) throws
NorthApiException
```


Class

Authentication

Parameter Description

AuthRefreshInDTO

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(256)	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
secret	Mandatory	String(256)	body	Indicates the password that corresponds to **appId**. It is used to log in to the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
refreshToken	Mandatory	String(256)	body	Indicates the refresh token used for obtaining a new **accessToken**. The value of this parameter is obtained when the **Auth** API is called.

Return Value

AuthRefreshOutDTO

Parameter	Type	Description
scope	String(256)	Indicates the applied permission range. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the access token type. This parameter has a fixed value of **bearer**.
expiresIn	Integer(256)	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.1.3 Periodically Refreshing a Token

Typical Scenario

An NA calls this API to periodically refresh the accessToken. The update interval is the value of **expiresIn** in the return value of the **Authentication** method to ensure that the accessToken does not expire. The accessToken is managed by the SDK. When calling other service APIs, the SDK can set the accessToken to **null**.

API Function

A new accessToken is automatically obtained before the existing accessToken expires.

Note

1. After this API is called, the SDK starts a thread to automatically refresh the accessToken. The accessToken is managed only in the SDK. NAs do not need to concern the accessToken. When calling other service APIs, the SDK sets the value of accessToken to **null**. If the value of accessToken is not **null**, the value of accessToken is used.
2. You can call the **stopRefreshTokenTimer()** method to stop a thread. After this method is called, the accessToken in a request to call another service API cannot be **null**. The accessToken is managed by NAs.

API Description

```
void startRefreshTokenTimer() throws NorthApiException
```

Class

Authentication

Parameter Description

The application ID and secret use the values of **ClientInfo** in the **NorthApiClient class**.

Return Value

void

2.3.1.4 Stopping Periodically Refreshing a Token

Typical Scenario

An NA calls this API to stop periodically refreshing the accessToken. Once the stopRefreshTokenTimer method is called, the accessToken cannot be **null**.

API Function

This API is used to stop automatic obtaining of a new accessToken and close the thread started by the stopRefreshTokenTimer method.

Note

After the stopRefreshTokenTimer method is call, the accessToken in the request for calling another service API cannot be **null**.

API Description

```
void stopRefreshTokenTimer()
```

Class

Authentication

Parameter Description

N/A

Return Value

void

2.3.2 Device Management

An NA adds a device to the IoT platform and obtains the device ID and verification code. After connecting to the IoT platform, the device establishes a subordinate relationship with the NA.

2.3.2.1 Registering a Device (Verification Code Mode)

Typical Scenario

Before a device accesses the IoT platform by using verification code, an NA needs to call this API to register the device on the IoT platform and set a unique identification code of the device (such as the IMEI) as the verification code. Then, the device can use the unique identification code to get authenticated and connect to the IoT platform.

API Function

This API is used by an NA to register a device with the IoT platform. After registration, the device can connect to the IoT platform.

API Description

```
RegDirectDeviceOutDTO regDirectDevice(RegDirectDeviceInDTO2 rddInDto, String
appId, String accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Parameter	Mandatory or Optional	Type	Location	Description
rddInDto	Mandatory	RegDirectDeviceInDTO2	body	For details, see **RegDirectDeviceInDTO2 structure**.

RegDirectDeviceInDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceInfo	Optional	DeviceInfoDTO	Body	Indicates information about the device. For details, see DeviceInfo structure.
imsi	Optional	String(1-64)	Body	Indicates the IMSI of an NB-IoT device.
isSecure	Optional	Boolean	Body	Indicates whether the device is secure. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure. **NOTE** If a user needs to register a secure device, this parameter must be specified.
verifyCode	Optional	String(256)	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. In the NB-IoT solution, this parameter is mandatory and must be set to the same value as **nodeId**.

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies the device. The value of this parameter must be the same as the device ID reported by the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
endUserId	Optional	String(256)	body	Identifies an end user. In the NB-IoT solution, this parameter is set to the IMSI of the device. In the SmartHome solution, this parameter is set to the application account.
psk	Optional	String(8-32)	body	If the pre-shared key (PSK) is specified in the request, the IoT platform uses the specified PSK. If the PSK is not specified in the request, the PSK is generated by the IoT platform. The value is a string of characters, including upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.

Parameter	Mandatory or Optional	Type	Location	Description
timeout	Optional	Integer(>=0)	Body	Indicates the validity period for device registration. When this API is called to register a device, the device can be bound within the validity period. If the device is not bound within the validity period, the registration information will be deleted. The value ranges from 0 to 2147483647. If this parameter is set to **0**, the device verification code is always valid. (The recommended value is **0**.) The default value is **180**. The default value can be configured. For details, contact the IoT platform maintenance personnel. Unit: second
productId	Optional	String(256)	Body	Identifies the product to which the device belongs.

DeviceInfoDTO:

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Optional	String(256)	Body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	Body	Indicates the manufacturer name.
deviceType	Optional	String(256)	Body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Mandatory or Optional	Type	Location	Description
model	Mandatory	String(256)	Body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
protocolType	Optional	String(256)	Body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Return Value

RegDirectDeviceOutDTO

Parameter	Type	Description
deviceId	String(256)	Identifies a device.
verifyCode	String(256)	Indicates the device verification code, with which the device can get authenticated and connect to the IoT platform. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
timeout	Integer	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period. (If this parameter is set to **0**, the verification code does not expire.)
psk	String(32)	Indicates a random PSK. If the PSK is carried in the request, it is used. Otherwise, the IoT platform generates a random PSK.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	103028	The license pool resources.	The license resources have been used up.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100416	The device has already been bounded.	The device has been bound to the IoT platform. Recommended handling: Check whether the device is registered.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application has not been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	600002	The product not existed.	The product does not exist. Recommended handling: Check whether **productId** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100412	The amount of device has reached the limit.	The number of devices under the current application reaches the upper limit. Recommended handling: Check whether the number of devices under the current application reaches the upper limit.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	103026	The license is not exist.	The license does not exist. Recommended handling: An internal license error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.2 Refreshing a Device Key

Typical Scenario

If the unique identification code of a device that has been registered with the IoT platform changes (for example, a device is replaced), an NA needs to call this API to update the unique identification code of the device and rebind the device.

NOTE

The device password can be updated only when the device is offline.

API Function

This API is used by an NA to update the node ID of a device that has been registered with the IoT platform, and rebind the device with the device ID unchanged.

API Description

```
RefreshDeviceKeyOutDTO refreshDeviceKey(RefreshDeviceKeyInDTO rdkInDTO, String
deviceId, String appId, String accessToken) throws NorthApiException
```


Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(256)	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.
rdkInDTO	Mandatory	RefreshDeviceKeyInDTO	body	For details, see **RefreshDeviceKeyInDTO structure**.

RefreshDeviceKeyInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
verifyCode	Optional	String(256)	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. You are advised to set this parameter to the value of **nodeId**.

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Optional	String(256)	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. ● If the value is **null**, the value of this parameter remains unchanged. ● If the value is not **null**, the value of this parameter is updated. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
timeout	Optional	Integer	body	Indicates the validity period of the verification code, in units of seconds. The value is an integer greater than or equal to **0**. ● If this parameter is set to **null**, the default value **180** prevails. ● If this parameter is set to **0**, the verification code never expires. ● If this parameter is not set to **0**, the verification code expires after the specified time elapses.

Return Value

RefreshDeviceKeyOutDTO structure

Parameter	Type	Description
verifyCode	String(256)	Indicates the device verification code, with which the device can get authenticated and connect to the IoT platform. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.

Parameter	Type	Description
timeout	Integer	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	100610	Device is not active.	The device has not been activated. Recommended handling: Check whether the device has been connected to the IoT platform and activated.
400	100611	Device is online.	The device is online. Recommended handling: Enable the device to go offline or disconnect the device from the IoT platform.

HTTP Status Code	Error Code	Error Description	Remarks
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.3 Modifying Device Information

Typical Scenario

After an NA registers a device on the IoT platform and the basic information about the device changes, the NA can also call this API to modify device information on the IoT platform.

API Function

This API is used to modify the basic information about a device, including the device type, device model, manufacturer, and access protocol.

API Description

```
void modifyDeviceInfo(ModifyDeviceInforInDTO mdiInDto, String deviceId, String
appId, String accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.
mdiInDto	Mandatory	ModifyDeviceInforInDTO	body	For details, see **ModifyDeviceInforInDTO structure**.

ModifyDeviceInforInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
customFields	Optional	List<**CustomField**>	Body	User-defined field list. Users can set customized fields.
imsi	Optional	String(1-64)	Body	Indicates the IMSI of an NB-IoT device.
name	Optional	String(1-256)	body	Indicates the device name.

Parameter	Mandatory or Optional	Type	Location	Description
endUser	Optional	String(1-256)	body	Indicates an end user. If the device is a directly connected device, this parameter is optional. If the device is a non-directly connected device, this parameter can be set to **null**.
mute	Optional	Enum	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
manufacturerId	Optional	String(1-256)	body	Uniquely identifies a manufacturer. After registering a device, you must change the manufacturer ID to be the same as that defined in the profile file.
manufacturerName	Optional	String(1-256)	body	Indicates the manufacturer name.
deviceType	Optional	String(1-256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**. In the NB-IoT solution, the device type must be changed to be the same as that defined in the profile after a device is registered.
model	Optional	String(1-256)	body	Indicates the device model, which is defined by the manufacturer. After registering a device, you must change the device model to be the same as that defined in the profile file.
location	Optional	String(1-1024)	body	Indicates the device location.

Parameter	Mandatory or Optional	Type	Location	Description
protocolType	Optional	String(1-256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**. After registering a device, you must change the protocol type to be the same as that defined in the profile file.
deviceConfig	Optional	DeviceConfigDTO	body	Indicates device configuration information. For details, see **DeviceConfigDTO structure**.
region	Optional	String(1-256)	body	Indicates the region information about the device.
organization	Optional	String(1-256)	body	Indicates the organization to which the device belongs.
timezone	Optional	String(1-256)	body	Indicates the time zone where the device is located. The time zone code is used. For example, the time zone code of Shanghai time zone is **Asia/Shanghai**.
ip	Optional	String(128)	Body	Indicates the device IP address.
isSecure	Optional	Boolean	body	Indicates the security status of the device. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure.
psk	Optional	String(8-32)	body	Indicates the PSK. The value is a string of characters that consist of only upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.
tags	Optional	List<**Tag2**>	Body	Indicates the tag of a device.

CustomField structure

Parameter	Mandatory or Optional	Type	Location	Description
fieldName	Optional	String(256)	Body	Indicates the field name.
fieldType	Optional	String(256)	Body	Indicates the field type.
fieldValue	Optional	String(256)	Body	Indicates the field value.

DeviceConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataConfig	Optional	DataConfigDTO	Indicates data configuration information. For details, see **DataConfigDTO structure**.

DataConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataAgingTime	Optional	Integer	Indicates the data aging time. The value range is 0-90. Unit: day.

Tag2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.
tagType	Optional	Integer	body	Indicates the tag type.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100440	The isSecure is invalid.	The value of **isSecure** is incorrect.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	500004	The amount of frozen devices has reached the limit.	The number of frozen devices has reached the upper limit.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.4 Deleting a Device

Typical Scenario

If a device that has been registered on the IoT platform does not need to connect to the platform, an NA can call this API to delete the device. If the device needs to connect to the IoT platform again, the NA must register the device again.

API Function

This API is used by an NA to delete a registered device from the IoT platform.

API Description

```
void deleteDirectDevice(String deviceId, Boolean cascade, String appId, String
accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
cascade	Mandatory	Boolean	query	This parameter is valid only when the device is connected to a non-directly connected device. If this parameter is not set, the value **null** is used. ● **true**: The directly connected device and the non-directly-connected devices connected to it are deleted. ● **false**: The directly connected device is deleted but the non-directly-connected devices connected to it are not deleted. In addition, the attribute of the non-directly-connected devices is changed to directly-connected.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.5 Querying Device Activation Status

Typical Scenario

After an NA registers a device on the IoT platform, the activation status of the device is **false** before the device connects to the IoT platform for the first time. When the device connects to the IoT platform for the first time, the activation status of the device is **true** regardless of whether the device is online, offline, or abnormal. The NA can call this API to query the activation status of the device to check whether the device has connected to the IoT platform.

API Function

This API is used by an NA to query the activation status of a device on the IoT platform to determine whether the device has connected to the IoT platform.

API Description

```
QueryDeviceStatusOutDTO queryDeviceStatus(String deviceId, String appId, String
accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Return Value

QueryDeviceStatusOutDTO

Parameter	Type	Description
deviceId	String(256)	Identifies a device.
activated	Boolean	Indicates whether the device is activated through a verification code. ● **true**: The device is activated. ● **false**: The device is not activated.
name	String(256)	Indicates the device name.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.6 Querying Device Shadow Information

Typical Scenario

When a device is in the offline or abnormal state, an NA cannot deliver configuration information to the device by sending a command. In this case, the NA can deliver the configuration information to the device shadow. When the device goes online, the device shadow will deliver the configuration information to the device. The NA can call this API to check the device configuration information and the latest data reported by the device on the device shadow.

API Function

This API is used by an NA to query the device shadow information of a device, including the device configuration information (in the desired section) and the latest data reported by the device (in the reported section).

API Description

```
QueryDeviceShadowOutDTO queryDeviceShadow(String deviceId, String appId, String
accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Return Value

QueryDeviceShadowOutDTO

Parameter	Type	Description
deviceId	String(36)	Identifies a device.
gatewayId	String(36)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Enum	Indicates the device type.
createTime	String(256)	Indicates the device creation time.
lastModifiedTime	String(256)	Indicates the last modification time.
deviceInfo	DeviceInfo	Indicates the device information. For details, see **DeviceInfo structure**.
services	List<DeviceServiceB>	Indicates service capabilities of the device. For details, see **DeviceServiceB structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely Identifies the manufacturer. Set this parameter to the value defined in the profile file of the device.
manufacturerName	String(256)	Indicates the manufacturer name. Set this parameter to the value defined in the profile file of the device.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used. Set this parameter to the value defined in the profile file of the device, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Type	Description
model	String(256)	Indicates the device model. Set this parameter to the value defined in the profile file of the device. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. Set this parameter to the value defined in the profile file of the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	String(256)	Indicates the device status details. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.

Parameter	Type	Description
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceServiceB structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
reportedProps	ObjectNode	Indicates the latest information reported by the device.
desiredProps	ObjectNode	Indicates the configuration information delivered to the device.
eventTime	String(256)	Indicates the time when an event occurs.
serviceType	String(256)	Indicates the service type.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.2.7 Modifying Device Shadow Information

Typical Scenario

The IoT platform supports the creation of device shadows. Device shadows store the latest service attribute data reported by devices and service attribute configurations delivered by NAs. (Service attributes are defined in the device profile file.) If the device is offline or abnormal, the NA cannot deliver configuration to the device by issuing commands. In this case, the NA can set the configuration to be delivered to the device shadow. When the device goes online again, the device shadow delivers the configuration to the device. The NA can call this API to modify the configuration information to be delivered to the device on the device shadow.

Each device has only one device shadow, which contains desired and report sections.

- The desired section stores the configurations of device service attributes. If a device is online, the configurations in the desired section are delivered to the device immediately. Otherwise, the configurations in the desired section are delivered to the device when the device goes online.
- The report section stores the latest service attribute data reported by devices. When a device reports data, the IoT platform synchronizes the data to the report section of the device shadow.

API Function

This API is used to modify the configuration information in the desired section of the device shadow. When the device goes online, the configuration information will be delivered to the device.

API Description

```
void modifyDeviceShadow(ModifyDeviceShadowInDTO mdsInDTO, String deviceId, String
appId, String accessToken) throws NorthApiException
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(256)	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.
mdsInDTO	Mandatory	ModifyDeviceShadowInDTO	body	For details, see **ModifyDeviceShadowInDTO structure**.

ModifyDeviceShadowInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceDesireds	Mandatory	List<ServiceDesiredDTO>	body	Indicates the configuration or status to be modified. For details, see **ServiceDesiredDTO structure**.

ServiceDesiredDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Optional	String(1-256)	body	Identifies a service.
desired	Optional	ObjectNode	body	Indicates the service attribute configuration of a device.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100425	The special deviceCapability is not exist.	The device template does not exist. Recommended handling: Check whether the device template has been uploaded to the IoT platform.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100443	The property is forbidden to write.	The device attributes cannot be written.

HTTP Status Code	Error Code	Error Description	Remarks
403	10100 09	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	10100 05	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	10040 3	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	10041 8	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	10002 3	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	10020 3	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.3 Batch Processing

NAs can perform batch operations on devices connected to the IoT platform through the batch processing API.

2.3.3.1 Creating a Batch Task

Typical Scenario

When an NA needs to perform an operation on a batch of devices, the NA can call this API to create a batch task. Currently, the supported batch operations include delivering pending commands to devices in batches.

API Function

This API is used by an NA to create a batch task for devices on the IoT platform.

API Description

```
BatchTaskCreateOutDTO createBatchTask(BatchTaskCreateInDTO2 btcInDTO, String
accessToken) throws NorthApiException
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.
btcInDTO	Mandatory	BatchTaskCreateInDTO2	body	For details, see **BatchTaskCreateInDTO2 structure**.

BatchTaskCreateInDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(64)	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
timeout	Mandatory	Integer	body	Indicates the timeout duration of a task, in minutes. The value range is 10 - 2880.
taskName	Mandatory	String	body	Indicates the task name. The value is a string of a maximum of 256 characters.
taskType	Mandatory	String	body	Indicates the task type. The value can be DeviceCmd.
param	Mandatory	ObjectNode	body	Indicates task parameters. The value of this parameter varies depending on the **taskType**. For details, see **ObjectNode structure**.
tags	Optional	List<TagDTO2>	body	Indicates the tag list. For details, see **TagDTO2 structure**.

ObjectNode:

Parameter	Mandatory or Optional	Type	Location	Description
type	mandatory	String	body	Indicates the batch command type. This parameter is mandatory when **taskType** is set to **DeviceCmd**. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	Conditionally mandatory	List<String>	body	Indicates the device ID list. This parameter is mandatory when **type** is set to **DeviceList**.
deviceType	Conditionally mandatory	String	body	Indicates the device type. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.

Parameter	Mandatory or Optional	Type	Location	Description
manufacturerId	Conditionally optional	String	body	Identifies the manufacturer. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
model	Conditionally optional	String	body	Indicates the device model. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
deviceLocation	Conditionally mandatory	String	body	Indicates the location of the device. This parameter is mandatory when **type** is set to **DeviceArea**.
groupList	Conditionally mandatory	List<String>	body	Indicates the group ID list or device group name list. When **type** is set to **GroupIdList**, set this parameter to the group ID. When **type** is set to **GroupList**, set this parameter to the device group name.
command	mandatory	**CommandDTO**	body	Indicates the command information.
callbackUrl	Conditionally optional	String	body	Indicates the push address of the command execution result.
maxRetransmit	Conditionally optional	Integer(0~3)	body	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	Conditionally optional	String	body	Indicates the group tag.

CommandDTO structure:

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	Mandatory	String(1-128)	body	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Optional	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.

Return Value

BatchTaskCreateOutDTO

Parameter	Type	Description
taskID	String	Identifies a batch task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	101001	Resource doesn't exist.	The resource does not exist.
200	105001	The batchTask count has reached the limit.	If the number of unfinished tasks is greater than or equal to 10, a message is returned, indicating that the number of tasks reaches the limit.
200	105002	The batchTask name has exist.	The task name exists. Recommended handling: Change the task name.
400	105201	The tagName and tagValue has been used on the platform.	The tagName and tagValue have been used on the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100028	The user has no right.	The user has no operation permission.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105202	The tag is not existed.	The tag does not exist.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.3.2 Querying Information About a Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about the batch task, including the task status and the subtask completion status.

API Function

This API is used by an NA to query the information about a single batch task by task ID.

API Description

```
QueryOneTaskOutDTO queryOneTask(String taskId, String select, String appId,
String accessToken) throws NorthApiException
```


Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
taskId	Mandatory	String	path	Identifies a batch task. The value of this parameter is obtained after the batch task is created.
select	Mandatory	String	query	Indicates an optional return value. The value can be **tag**. If this parameter is not specified, the value can be **null**.
appId	Mandatory	String	query	If the task belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Return Value

QueryOneTaskOutDTO

Parameter	Type	Description
appId	String	Indicates the **appId** to which the batch task belongs.
taskId	String	Identifies a batch task.
taskName	String	Indicates the name of the batch task.
operator	String	Indicates the operator who delivers the batch task.
taskFrom	String	Indicates the source of the batch task. ● **Portal**: The task is created on the SP portal. ● **Northbound**: The task is created by calling a northbound API.
taskType	String	Indicates the type of the batch task. The value can be DeviceCmd.
status	String	Indicates the status of the batch task. The value options are **Pending**, **Running**, **Complete**, and **Timeout**.

Parameter	Type	Description
startTime	String	Indicates the time when the batch task is created.
timeout	Integer	Indicates the time when the batch task times out, in seconds.
progress	Integer	Indicates the progress of the batch task. The value is a permillage ranging from 0 to 1000 and is rounded down.
totalCnt	Integer	Indicates the total number of tasks.
successCnt	Integer	Indicates the number of tasks that are successfully executed.
failCnt	Integer	Indicates the number of tasks that fail to be executed.
timeoutCnt	Integer	Indicates the number of tasks with execution timeout.
expiredCnt	Integer	Indicates the number of expired tasks that are not executed.
completeCnt	Integer	Indicates the number of completed tasks, including successful, failed, and timed out tasks.
successRate	Integer	Indicates the task success rate. The value is a permillage ranging from 0 to 1000 and is rounded down.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.
tags	List<**TagDTO2**>	Indicates the tag list of the batch task.

ObjectNode structure:

Parameter	Type	Description
type	String	Indicates the batch command type. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	List<String>	Indicates the device ID list. The value is that returned when **type** is set to **DeviceList**.
deviceType	String	Indicates the type of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
manufacturerId	String	Identifies a manufacturer. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.

Parameter	Type	Description
model	String	Indicates the model of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
deviceLocation	String	Indicates the location of the device. The value is that returned when **type** is set to **DeviceArea**.
groupList	List<String>	Indicates the group name list. The value is that returned when **type** is set to **GroupList**.
command	**CommandDTO**	Indicates the command information. .
callbackUrl	String	Indicates the push address of the command execution result..
maxRetransmit	Integer(0~3)	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	String	Indicates the group tag.

CommandDTO structure:

Parameter	Type	Description
serviceId	String(1-64)	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure:

Parameter	Type	Description
tagName	String(1-128)	Indicates the tag name.
tagValue	String(1-1024)	Indicates the tag value.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100019	Illegal request.	Invalid request. Recommended handling: Check whether the mandatory parameters in the request are set.
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

2.3.3.3 Querying Information About a Subtask of a Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about a subtask of the batch task, including the subtask execution status and subtask content.

API Function

This API is used by an NA to query information about a subtask of a batch task based on specified conditions. This API applies to the current application.

API Description

```
QueryTaskDetailsOutDTO queryTaskDetails(QueryTaskDetailsInDTO qtdInDTO, String
accessToken) throws NorthApiException
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.
qtdInDTO	Mandatory	QueryTaskDetailsInDTO	query	For details, see **QueryTaskDetailsInDTO structure**.

QueryTaskDetailsInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform. Set this parameter to the value of **appId** of the authorized application.
taskId	Mandatory	String	query	Identifies a batch task.
status	Optional	String	query	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
deviceId	Optional	String	query	Identifies a device. This parameter is used for querying details about a batch command delivery task.
commandId	Optional	String	query	Identifies a command. This parameter is used for querying details about a batch command delivery task.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.

Return Value

QueryTaskDetailsOutDTO

Parameter	Type	Description
pageNo	Integer	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.

Parameter	Type	Description
pageSize	Integer	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.
totalCount	Integer	Indicates the total number of records.
taskDetails	List<QueryTaskDetailDTOCloud2NA>	Indicates the task details list. For details, see **QueryTaskDetailDTOCloud2NA structure**.

QueryTaskDetailDTOCloud2NA structure

Parameter	Type	Description
status	String	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
output	String	Indicates the output of a batch command delivery task.
error	String	Indicates the cause of error, in the format of {\"error_code\": \"****\", \"error_desc\":\"*****\"}.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.

ObjectNode structure:

Parameter	Type	Description
deviceId	String	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration..
commandId	String	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.4 Subscription Management

The IoT platform allows NAs to subscribe to device data. If the subscribed device data changes, the IoT platform pushes change notifications to NAs. The subscription management API must be used together with the **Message Push** API.

2.3.4.1 Subscribing to Service Data of the IoT Platform

Typical Scenario

An NA can subscribe to service data of a device on the IoT platform. When the service data changes (for example, the device is registered, the device reports data and the device status changes), the IoT platform can push change notifications to the NA. The NA can call this API to subscribe to different types of service change notifications.

API Function

This API is used by an NA to subscribe to service change notifications on the IoT platform. When the device status or data changes, the IoT platform pushes notifications to the NA.

API Description

```
SubscriptionDTO subDeviceData(SubDeviceDataInDTO sddInDTO, String ownerFlag,
String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
sddInDTO	Mandatory	SubDeviceDataInDTO structure	body	For details, see SubDeviceDataInDTO structure.

Parameter	Mandatory or Optional	Type	Location	Description
ownerFlag	Mandatory	String(256)	query	Identifies the owner of the callbackUrl. If this parameter is not specified, this parameter can be set to **null**. ● **false**: The callback URL owner is an authorizing application. ● **true**: The callback URL owner is an authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

SubDeviceDataInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceAdded**: device addition; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription ● **commandRsp**: command response; sending such a notification after subscription ● **deviceEvent**: device event; sending such a notification after subscription ● **serviceInfoChanged**: service information change; sending such a notification after subscription ● **deviceModelAdded**: device model addition; sending such a notification after subscription ● **deviceModelDeleted**: device model deletion; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change; sending such a notification after subscription

Parameter	Mandatory or Optional	Type	Location	Description
callbackUrl	Mandatory	String(1024)	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.
appId	Optional	String(256)	body	Identifies the application of the entity that subscribes to a device or rule.
channel	Optional	String(32)	Body	Indicates the transmission channel. For the MQTT client, the value is **MQTT**. In other cases, the value is **HTTP**.

Response Parameters

SubscriptionDTO

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.
clientIds	List<String>	Identifies an MQTT client. The value is that returned when MQTT is subscribed to.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	The request callbackurl is illegal.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
409	100227	The resource is conflicted.	A resource conflict occurs. The notification type has been subscribed to. Recommended handling: Check whether the notification type has been subscribed.

2.3.4.2 Subscribing to Management Data of the IoT Platform

Typical Scenario

An NA can subscribe to management data of a device on the IoT platform. When operations are performed on the device (for example, device upgrade), the IoT platform notifies the NA of the operation status or results. The NA can call this API to subscribe to different types of device upgrade notifications on the IoT platform.

API Function

This API is used by an NA to subscribe to device upgrade notifications on the IoT platform. When the device is upgraded, the IoT platform sends a notification to the NA.

API Description

```
void subDeviceData(SubDeviceManagementDataInDTO smdInDTO, String accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
smdInDTO	Mandatory	SubDeviceManagementDataInDTO structure	body	For details, see SubDeviceManagementDataInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

SubDeviceManagementDataInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. ● **swUpgradeStateChangeNotify**: software upgrade status change notification; sending such a notification after subscription ● **swUpgradeResultNotify**: software upgrade result notification; sending such a notification after subscription ● **fwUpgradeStateChangeNotify**: hardware upgrade status change notification; sending such a notification after subscription ● **fwUpgradeResultNotify**: hardware upgrade result notification; sending such a notification after subscription

Parameter	Mandatory or Optional	Type	Location	Description
callbackurl	Mandatory	String	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	Internal server error.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
400	100228	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100229	Get AppKey from header failed.	Failed to obtain the AppKey from the message header.
500	100244	register out route fail.	Failed to register the route. Recommendation: Contact the IoT platform maintenance personnel.

2.3.4.3 Querying a Subscription

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query configuration information about a subscription.

API Function

This API is used by an NA to query the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
SubscriptionDTO querySingleSubscription(String subscriptionId, String appId,
String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
subscriptionId	Mandatory	String	path	Identifies a subscription, which is obtained by calling or querying the subscription API.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	Identifies the application of the entity that subscribes to a device or rule.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

SubscriptionDTO

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

2.3.4.4 Querying Subscription in Batches

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query all subscription configurations of the current application or of a specified subscription type.

API Function

This API is used to query all subscription information of the current application or of a specified subscription type.

API Description

```
QueryBatchSubOutDTO queryBatchSubscriptions(QueryBatchSubInDTO qbsInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qbsInDTO	Mandatory	QueryBatchSubInDTO structure	query	For details, see QueryBatchSubInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryBatchSubInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String(256)	query	Identifies the application of the entity that subscribes to a device or rule.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted**: device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the page size. The value is an integer greater than or equal to **1**. The default value is **10**.

Response Parameters

QueryBatchSubOutDTO

Parameter	Type	Description
totalCount	long	Indicates the total number of records.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records on each page.
subscriptions	List<SubscriptionDTO>	Indicates the subscription information list. For details, see **SubscriptionDTO structure**.

SubscriptionDTO structure

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100224	The resource exceeds 1000, please refinement query conditions.	The number of resources exceeds 1000. Recommended handling: Narrow down the filter criteria.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

2.3.4.5 Deleting a Subscription

Typical Scenario

If an NA does not need to receive a subscription notification message pushed by the IoT platform, the NA can call this API to delete the specified subscription configuration and cancel the subscription.

API Function

This API is used by an NA to delete the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
void deleteSingleSubscription(String subscriptionId, String appId, String
accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
subscriptionId	Mandatory	String(256)	path	Identifies a subscription.
appId	Optional	String(256)	query	Identifies the application of the entity that subscribes to a device or rule.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **subscriptionId** is correct.

2.3.4.6 Deleting Subscriptions in Batches

Typical Scenario

If an NA does not need to receive subscription notification messages pushed by the IoT platform or a specified type of subscription notification messages, the NA can call this API to delete subscription configurations in batches and cancel the subscriptions.

API Function

This API is used to delete all subscriptions, subscriptions of a specified subscription type, or subscriptions of a specified callback URL in batches.

API Description

```
void deleteBatchSubscriptions(DeleteBatchSubInDTO dbsInDTO, String accessToken)
throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dbsInDTO	Mandatory	DeleteBatchSubInDTO	body	For details, see the DeleteBatchSubInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

DeleteBatchSubInDTO

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String(256)	query	Identifies the application of the entity that subscribes to a device or rule.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted**: device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification
callbackUrl	Optional	String(256)	query	Indicates the callback URL of the subscription.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **notifyType** is correct.

2.3.5 Message Push

NAs can subscribe to device information from the IoT platform. When the device information changes, the IoT platform pushes change notifications to the NAs. Then, the NAs distribute messages based on the notification type. This API must be used together with the **Subscription Management** API.

2.3.5.1 Pushing Device Registration Notifications

Typical Scenario

After an NA subscribes to device registration notifications (the notification type is **deviceAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when the NA registers a device on the IoT platform by calling the API for registering a directly connected device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device registration notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceAdded(NotifyDeviceAddedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceAddedDTO

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceAdded**.
deviceId	Mandatory	String	body	Identifies a device.
gatewayId	Optional	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Mandatory	String	body	Indicates the device type. ● ENDPOINT ● GATEWAY ● UNKNOWN
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.

Parameter	Mandatory or Optional	Type	Location	Description
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	Optional	String(256)	body	Indicates status details about the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.

Parameter	Mandatory or Optional	Type	Location	Description
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceAdded",
    "deviceId":"*****",
    "gatewayId":"*****",
    "nodeType":"GATEWAY",
    "deviceInfo":{
        "nodeId":"******",
        "name":null,
        "description":null,
        "manufacturerId":null,
        "manufacturerName":null,
        "mac":null,
        "location":null,
        "deviceType":null,
        "model":null,
        "swVersion":null,
```



```
        "fwVersion":null,
        "hwVersion":null,
        "protocolType":null,
        "bridgeId":null,
        "status":"OFFLINE",
        "statusDetail":"NOT_ACTIVE",
        "mute":null,
        "supportedSecurity":null,
        "isSecurity":null,
        "signalStrength":null,
        "sigVersion":null,
        "serialNumber":null,
        "batteryLevel":null
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.2 Pushing Device Binding Notifications

Typical Scenario

After an NA subscribes to device binding notifications (the notification type is **bindDevice**) on the IoT platform, the IoT platform sends a notification message to the NA when a directly connected device is connected to the IoT platform and bound to the NA.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device binding notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleBindDevice(NotifyBindDeviceDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyBindDeviceDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **bindDevice**.
deviceId	Mandatory	String	body	Identifies a device.
resultCode	Mandatory	String	body	Indicates the binding result. The value options are **expired** and **succeeded**.
deviceInfo	Optional	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Identifies a device.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Mandatory or Optional	Type	Location	Description
model	Optional	String(256)	body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **Z-Wave**, **ZigBee**, and **WPS**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	Optional	String(256)	body	Indicates the device status details. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Mandatory or Optional	Type	Location	Description
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
```



```
Header:
Content-Type:application/json
Body:
{
    "notifyType":"bindDevice",
    "deviceId":"*****",
    "resultCode":"succeeded",
    "deviceInfo":{
        "name":"Sensor_12",
        "manufacturer":"wulian",
        "deviceType":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
         "fwVersion": "...",
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smockdetector",
        "nodeType":"GATEWAY"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.3 Pushing Device Information Change Notifications

Typical Scenario

After an NA subscribes to device information change notifications (the notification type is **deviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device configuration or status (such as manufacturer, location, version and online status) changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device information change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceInfoChanged(NotifyDeviceInfoChangedDTO body)

Class	PushMessageReceiver

Parameter Description

NotifyDeviceInfoChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceInfoChanged**.
deviceId	Mandatory	String	body	Identifies a device.
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Identifies a device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.

Parameter	Mandatory or Optional	Type	Location	Description
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.

Parameter	Mandatory or Optional	Type	Location	Description
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	Optional	String(256)	body	Indicates the device status details. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType ":"deviceInfoChanged",
    "deviceId":"******************************",
    "gatewayId":"*************",
    "deviceInfo":{
        "name":"Sensor_12",
        "manufacturer":"wulian",
        "type":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
         "fwVersion": "...",
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smock detector"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.4 Pushing Device Data Change Notifications

Typical Scenario

After an NA subscribes to device data change notifications (the notification type is **deviceDataChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of a single service attribute.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device data change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceDataChanged(NotifyDeviceDataChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDataChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDataChanged**.
requestId	Optional	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
service	Mandatory	DeviceService	body	Indicates service data of the device. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.
eventTime	Mandatory	String	body	Indicates the time when an event occurs. The value is in the format of yyyymmddThhmmssZ. An example value is **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDataChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":{
        "serviceId":"Brightness",
        "serviceType":"Brightness",
        "data":{
            "brightness":80
        },
        "eventTime":"20170311T163657Z"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.5 Pushing Batch Device Data Change Notifications

Typical Scenario

After an NA subscribes to batch device data change notifications (the notification type is **deviceDatasChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of multiple service attributes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to batch device data change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceDatasChanged(NotifyDeviceDatasChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDatasChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDataChanged**.
requestId	Optional	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
services	Mandatory	List<DeviceService>	body	Indicates a service list. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.
eventTime	Mandatory	String	body	Indicates the time when an event is reported. The value is in the format of yyyymmddThhmmssZ. An example value is **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDatasChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":[
        {
            "serviceId":"Brightness",
            "serviceType":"Brightness",
            "data":{
                "brightness":80
            },
            "eventTime":"20170311T163657Z"
        },
        {
            "serviceId":"Color",
            "serviceType":"Color",
            "data":{
                "value":"red"
            },
            "eventTime":"20170311T163657Z"
        }
    ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```


2.3.5.6 Pushing Device Service Information Change Notifications

Typical Scenario

After an NA subscribes to device service information change notifications (the notification type is **serviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device to modify the device service information.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service information change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleServiceInfoChanged(NotifyServiceInfoChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyServiceInfoChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	Enum	body	Indicates the notification type. The value is **serviceInfoChanged**.
deviceId	Mandatory	String	body	Identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
serviceInfo	Mandatory	ServiceInfo	body	Indicates the masked device service information, which is incrementally reported. For details, see **ServiceInfo structure**.

ServiceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
muteCmds	Optional	List<String>	body	Indicates the device command list.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"serviceInfoChanged",
    "deviceId":"*******",
    "serviceId":"*******",
    "serviceType":"*******",
    "gatewayId":"*******",
    "serviceInfo":{
        "muteCmds":"VIDEO_RECORD"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.7 Pushing Device Deletion Notifications

Typical Scenario

After an NA subscribes to device deletion notifications (the notification type is **deviceDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when the device is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device deletion notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceDeleted(NotifyDeviceDeletedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDeletedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDeleted**.
deviceId	Mandatory	String	body	Identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Mandatory	String	body	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDeleted",
    "deviceId":"********************************",
    "gatewayId":"*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.8 Pushing Device Acknowledgment Notifications

Typical Scenario

After an NA subscribes to device acknowledgment notifications (the notification type is **messageConfirm**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command acknowledgment message (for example, the command is delivered or executed).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device acknowledgment notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleMessageConfirm(NotifyMessageConfirmDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyMessageConfirmDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	header	Indicates the notification type. The value is **messageConfirm**.
header	Mandatory	MessageConfirmHeader	header	For details, see **MessageConfirmHeader structure**.
body	Mandatory	ObjectNode	body	Based on the service definition, an acknowledgment message can carry information such as status change.

MessageConfirmHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.

Parameter	Mandatory or Optional	Type	Location	Description
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: /devices/{deviceId} ● Request initiated by a device service: /devices/{deviceId}/services/{serviceId}
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
status	Mandatory	String(1-32)	body	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been received. ● **executed**: The command has been executed.
timestamp	Mandatory	String(1-32)	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
"notifyType": "messageConfirm",
"header": {
"requestId": "************",
"from": "************",
"to": "************",
"status": "delivered",
"timestamp": "20151212T121212Z"
},
"body": {
}
}
```


Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.9 Pushing Device Command Response Notifications

Typical Scenario

After an NA subscribes to device command response notifications (the notification type is **commandRsp**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command response message (for example, the command execution succeeds or fails).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device command response notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleCommandRsp(NotifyCommandRspDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyCommandRspDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **commandRsp**.

Parameter	Mandatory or Optional	Type	Location	Description
header	Mandatory	CommandRspHeader	body	For details, see **CommandRspHeader structure**.
body	Mandatory	ObjectNode	body	Indicates the content of the message body of the response command.

CommandRspHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: /devices/{deviceId} ● Request initiated by a device service: /devices/{deviceId}/services/{serviceId}
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
deviceId	Mandatory	String	body	Identifies a device.
serviceType	Mandatory	String	body	Indicates the type of the service to which a command belongs.
method	Mandatory	String(1-128)	body	Indicates a stored response command, for example, **INVITE-RSP**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
```

```
Header:
Content-Type:application/json
Body:
{
    "notifyType":"commandRsp",
    "header":{
        "requestId":"***********",
        "from":"***********",
        "to":"***********",
        "deviceId":"***********",
        "serviceType":"Camera",
        "method":"MUTE_COMMANDS"
    },
    "body":{

    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.10 Pushing Device Event Notifications

Typical Scenario

After an NA subscribes to device event notifications (the notification type is **deviceEvent**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform receives the event message reported by the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device event notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceEvent(NotifyDeviceEventDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceEventDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceEvent**.
header	Mandatory	DeviceEventHeader	body	For details, see **DeviceEventHeader structure**.
body	Mandatory	Object Node	body	Indicates the content of the event message.

DeviceEventHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
eventType	Mandatory	String(1-32)	body	Indicates the event type.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: /devices/{deviceId} ● Request initiated by a device service: /devices/{deviceId}/services/{serviceId}
timestamp	Mandatory	String (1-32)	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
eventTime	Mandatory	String (1-32)	body	Indicates the time when an event is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
deviceId	Mandatory	String	body	Identifies a device.
serviceType	Mandatory	String	body	Indicates the type of the service to which a command belongs.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceEvent",
    "header":{
        "eventType":"*******",
        "from":"/devices/{deviceId}/services/{serviceId}",
        "deviceId":"*******",
        "serviceType":"*******",
        "timestamp":"20151212T121212Z",
        "eventTime":"20151212T121212Z"
    },
    "body":{
        "usedPercent":80
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.11 Pushing Device Model Addition Notifications

Typical Scenario

After an NA subscribes to device model addition notifications (the notification type is **deviceModelAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is added on the IoT platform

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model addition notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceModelAdded(NotifyDeviceModelAddedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceModelAddedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceModelAdded**.
appId	Mandatory	String	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the name of the manufacturer of the device model.
manufacturerId	Mandatory	String	body	Identifies the manufacturer of the device model.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
```



```
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceType":"*************",
    "manufacturerName":"wulian",
    " manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"zigbee"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.12 Pushing Device Model Deletion Notifications

Typical Scenario

After an NA subscribes to device model deletion notifications (the notification type is **deviceModelDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model deletion notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceModelDeleted(NotifyDeviceModelDeletedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceModelDeletedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceModelDeleted**.
appId	Mandatory	String	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the name of the manufacturer of the device model.
manufacturerId	Mandatory	String	body	Identifies the manufacturer of the device model.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceTyp ":"*************",
    " manufacturerName":"*************",
    "manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


2.3.5.13 Pushing Device Shadow Status Change Notifications

Typical Scenario

After an NA subscribes to device shadow status change notifications (the notification type is **deviceDesiredPropertiesModifyStatusChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device shadow on the IoT platform succeeds or fails to synchronize data to the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device shadow status change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleDeviceDesiredStatusChanged(NotifyDeviceDesiredStatusChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDesiredStatusChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDesiredPropertiesModifyStatusChanged**.
deviceId	Mandatory	String	body	Identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
properties	Mandatory	ObjectNode	body	Indicates data attributes of a device shadow.
status	Mandatory	String	body	Indicates the status. The value options are **DELIVERED** and **FAILED**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDesiredPropertiesModifyStatusChanged",
    "deviceId":"******",
    "serviceId":"Device",
    "properties":{
        "Model Number":1,
        "Serial Number":2,
        "Firmware Version":"v1.1.0"
    },
    "status":"DELIVERED"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.14 Pushing Software Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to software upgrade status change notifications (the notification type is **swUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the software upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade status change notifications.

Note

1. When **subscribing to platform management data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleSwUpgradeStateChanged(NotifySwUpgradeStateChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifySwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeStateChangeNotify**.
deviceId	Mandatory	String	body	Identifies a device.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade subtask.

Parameter	Mandatory or Optional	Type	Location	Description
swUpgradeState	Mandatory	String	body	Indicates the software upgrade status. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"swUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"***********",
    "subOperationId":"***********",
    "swUpgradeState":"downloading"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.15 Pushing Software Upgrade Result Notifications

Typical Scenario

After an NA subscribes to software upgrade result change notifications (the notification type is **swUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a software upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade result notifications.

Note

1. When **subscribing to platform management data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleSwUpgradeResult(NotifySwUpgradeResultDTO body)
Class	PushMessageReceiver

Parameter Description

NotifySwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeResultNotify**.
deviceId	Mandatory	String	body	Identifies a device.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current software version of the device.
targetVersion	Mandatory	String	body	Indicates the target software version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source software version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
swUpgradeResult	Mandatory	String	body	Indicates the software upgrade result. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the upgrade time.
resultDesc	Mandatory	String	body	Indicates the upgrade result description.
errorCode	Mandatory	String	body	Indicates a status error code reported by the device.
description	Mandatory	String	body	Indicates the description of the cause of error.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"swUpgradeResultNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"***********",
    "subOperationId":"***********",
    "curVersion":"1.3",
    "targetVersion":"1.5",
    "sourceVersion":"1.0",
    "swUpgradeResult":"SUCCESS",
    "upgradeTime":"***",
    "resultDesc":"***",
    "errorCode":"***",
    "description":"***"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


2.3.5.16 Pushing Firmware Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to firmware upgrade status change notifications (the notification type is **fwUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the firmware upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade status change notifications.

Note

1. When **subscribing to platform management data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleFwUpgradeStateChanged(NotifyFwUpgradeStateChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyFwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeStateChangeNotify**.
deviceId	Mandatory	String	body	Identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade sub-task.
step	Mandatory	String	body	Indicates the firmware upgrade status. The value options are **0**, **1**, **2**, and **3**.
stepDesc	Mandatory	String	body	Indicates the upgrade status description. ● **downloading** (corresponding to value **1** of **step**): The device is downloading the firmware package. ● **downloaded** (corresponding to value **2** of **step**): The device has finished downloading the firmware package. ● **updating** (corresponding to value **3** of **step**): The device is being upgraded. ● **idle** (corresponding to value **0** of **step**): The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"**************",
    "subOperationId":"***********",
    "step":"1",
    "stepDesc":"downloading"
}
```


Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.17 Pushing Firmware Upgrade Result Notifications

Typical Scenario

After an NA subscribes to firmware upgrade result change notifications (the notification type is **fwUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a firmware upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade result notifications.

Note

1. When **subscribing to platform management data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	void handleFwUpgradeResult(NotifyFwUpgradeResultDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyFwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeResultNotify**.
deviceId	Mandatory	String	body	Identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current firmware version of the device.
targetVersion	Mandatory	String	body	Indicates the target firmware version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source firmware version of the device.
status	Mandatory	String	body	Indicates the upgrade result. ● SUCCESS ● FAIL
statusDesc	Mandatory	String	body	Indicates the upgrade result description. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the firmware upgrade duration.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeResultNotify",
    "deviceId":"******",
    "appId":"**********",
    "operationId":"**********",
    "subOperationId":"**********",
    "curVersion":"1.6",
    "targetVersion":"1.6",
    "sourceVersion":"1.3",
    "status":"SUCCESS",
    "statusDesc":"****",
    "upgradeTime":"****"
}
```


Response Example

```
Response:
Status Code: 200 OK
```

2.3.5.18 Pushing NB-IoT Command Status Change Notifications

Typical Scenario

When an NA creates a device command with the callback URL specified, the IoT platform pushes a notification message to the NA if the command status changes (failed, successful, timeout, sent, or delivered).

API Function

The IoT platform pushes notification messages to NAs when the command status changes.

Note

1. When **creating an NB-IoT device command**, an NA must set the callback address in the API description. The server and port in the callback address are the public IP address of the NA and the port specified in the server configuration.

2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver/cmd
Callback API	void handleNBCommandStateChanged(NotifyNBCommandStatusChangedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyNBCommandStatusChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	body	Uniquely identifies a device.
commandId	Mandatory	String	body	Identifies a command, which is generated by the IoT platform during device creation.

Parameter	Mandatory or Optional	Type	Location	Description
result	Mandatory	NBCommandResult	body	For details, see **NBCommandResult structure**.

NBCommandResult structure

Parameter	Mandatory or Optional	Type	Location	Description
resultCode	Mandatory	String	body	Indicates the command status result. ● **SENT**: The IoT platform has delivered a command to the device but has not received a response from the device. ● **DELIVERED**: The IoT platform receives a response from a device. ● **SUCCESS**: The IoT platform receives a command result and the result is success. ● **FAIL**: The IoT platform receives a command result and the result is a failure.
resultDetail	Mandatory	Object Node	body	Indicates the user-defined fields carried in the command result.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "deviceId":"*********",
    "commandId":"***********",
    "result":{
        "resultCode":"DELIVERED",
        "resultDetail":null
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

2.3.6 Command Delivery (NB-IoT Commands)

2.3.6.1 Creating Device Commands

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform provides two command delivery modes:

- Immediate delivery: The IoT platform delivers commands to devices immediately after receiving the commands. This ensures real-time performance but does not ensure serialization.
- Pending delivery: After receiving commands, the IoT platform caches the commands. When the devices are reachable, the IoT platform delivers the commands in sequence. Specifically, the IoT platform delivers the latter command only after receiving the response of the previous command (which is the ACK automatically replied by the module) to ensure serialization instead of real-time performance.

API Function

This API is used by NAs to deliver commands to devices. Immediate delivery and pending delivery are supported on the IoT platform.

API Description

```
PostDeviceCommandOutDTO2 postDeviceCommand(PostDeviceCommandInDTO2 pdcInDTO,
String appId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
pdcInDTO	Mandatory	PostDeviceCommandInDTO2 structure	body	For details, see PostDeviceCommandInDTO2 structure.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

PostDeviceCommandInDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(64)	body	Uniquely identifies the device that delivers the command.
command	Mandatory	CommandDTOV4	body	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	Optional	String(1024)	body	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Optional	Integer(>=0)	body	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2). If this parameter is set to **0**, the IoT platform will deliver the command to the specific device immediately regardless of the command mode set on the IoT platform (if the device is sleeping or the link has aged, the device cannot receive the command, the IoT platform cannot receive any response from the device, and the command times out in the end).

Parameter	Mandatory or Optional	Type	Location	Description
maxRetransmit	Optional	Integer(0-3)	body	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as **serviceId** defined in the profile.
method	Mandatory	String(1-128)	body	Indicates the command name. The value of this parameter must be the same as the command name defined in the profile file.
paras	Mandatory	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device. If no parameter is defined in the command in the profile file, left it blank, that is, "paras": {}.

Response Parameters

PostDeviceCommandOutDTO2 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.

Parameter	Type	Description
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **PENDING**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command can be retransmitted.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
400	100223	Command counts has reached the upLimit.	The number of cached commands reaches the limit. The number of commands in the **PENDING** state does not exceed the limit. The default value is **20**. Recommended handling: If the commands cached on the IoT platform need to be executed, enable the device to report data to trigger delivery of the cache commands. If a command cached on the IoT platform does not need to be executed, call the API used for modifying device commands V4 to change the state of the command from **PENDING** to **CANCELED**.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100612	Device is zombie.	The device is a zombie device. (The interval between the current system time and the time when the device went online exceeds the threshold. The default value is seven days.) Recommended handling: Run the command again after the device goes online.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.6.2 Querying Device Commands

Typical Scenario

After an NA delivers a command to a device, the NA can call this API to query the status and content of the delivered command on the IoT platform to check the command execution status.

API Function

This API is used by an NA to query the status and content of delivered commands on the IoT platform. All the commands delivered by the current application in a specified period or all the commands delivered to a specified device can be queried.

API Description

```
QueryDeviceCommandOutDTO2 queryDeviceCommand(QueryDeviceCommandInDTO2 qdcInDTO,
String accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcInDTO	Mandatory	QueryDeviceCommandInDTO2 structure	query	For details, see QueryDeviceCommandInDTO2 structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceCommandInDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
pageSize	Optional	Integer(>=1&&<=1000)	query	Indicates the number of records to be displayed on each page. The value ranges from 1 to 1000. The default value is **1000**.
deviceId	Optional	String(64)	query	Identifies the device whose commands are to be queried.
startTime	Optional	String	query	Indicates the start time. Commands delivered later than the start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Commands delivered earlier than the end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCommandOutDTO2 structure

Parameter	Type	Description
pagination	Pagination	Indicates pagination information. For details, see **Pagination structure**.
data	List<DeviceCommandRespV4>	Indicates the device command list. For details, see **DeviceCommandRespV4 structure**.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalSize	long	Indicates the total number of records.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.

Parameter	Type	Description
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **PENDING**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates the command parameter, which is a JSON string. The specific format is negotiated by the NA and device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: ● Ensure that neither **startTime** nor **endTime** is null and the value of **endTime** is later than that of **startTime**. ● Ensure that **pageNo** is not null and the value of **pageNo** is greater than **0**. ● Ensure that **pageSize** is not null and the value of **pageSize** is greater than **1**.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.6.3 Modifying Device Commands

Typical Scenario

NAs can call this API to modify the status of commands that have not been canceled, expired, or executed. Currently, the status of such commands can only be changed to **Canceled**.

API Function

This API is used by NAs to modify the status of commands. Currently, the status of such commands can only be changed to **Canceled**. That is, the commands are revoked.

API Description

```
UpdateDeviceCommandOutDTO updateDeviceCommand(UpdateDeviceCommandInDTO udcInDTO,
String deviceCommandId, String appId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
udcInDTO	Mandatory	UpdateDeviceCommandInDTO structure	body	For details, see UpdateDeviceCommandInDTO structure.
deviceCommandId	Mandatory	String	path	Identifies the command whose status is to be modified. The value of this parameter is obtained after the API used for creating device commands is called.
appId	Mandatory	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

UpdateDeviceCommandInDTO structure

Parameter	Mandatory or Optional	Type	Description
status	Mandatory	String	Indicates the command execution result. The value can be **CANCELED**, which indicates that the command is revoked.

Response Parameters

UpdateDeviceCommandOutDTO structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **PENDING**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled. ● **DELIVERED**: The command has been delivered. ● **SENT**: The command is being delivered.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.

Parameter	Type	Description
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	body	Indicates the command name.
paras	Optional	Object	body	Indicates the command parameter, which is a JSON string. The specific format is negotiated by the NA and device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
200	100434	The device command is not existed.	The device command does not exist. Recommended handling: Check whether the device command ID in the request is correct.
200	100435	The device command already canceled, expired or executed, Cannot cancel.	The device command has been canceled, expired, or executed. It cannot be canceled.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.6.4 Creating Device Command Revocation Tasks

Typical Scenario

After an NA delivers commands to a device, the IoT platform does not deliver the commands to the device for execution (the commands are in the **DEFAULT** state) if the commands are in queue or the device is offline. In this case, the NA can call this API to revoke all the undelivered commands of a specified device. Commands that have been delivered cannot be revoked.

API Function

This API is used by an NA to create a command revocation task to revoke all undelivered commands (that is, commands in the **DEFAULT** state) with the specified device ID on the IoT platform.

API Description

```
CreateDeviceCmdCancelTaskOutDTO
createDeviceCmdCancelTask(CreateDeviceCmdCancelTaskInDTO cdcctInDTO, String
appId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cdcctInDTO	Mandatory	CreateDeviceCmdCancelTaskInDTO structure	body	For details, see CreateDeviceCmdCancelTaskInDTO structure.
appId	Mandatory	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

CreateDeviceCmdCancelTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(1-64)	body	Identifies the device whose commands are to be revoked. The revocation task will revoke all commands delivered to this device.

Response Parameters

CreateDeviceCmdCancelTaskOutDTO structure

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Identifies the device whose commands are to be revoked by the revocation task.
status	String	Indicates the status of the command revocation task. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<DeviceCommandRespV4>	Indicates the revoked device command list. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.

Parameter	Type	Description
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates the command parameter, which is a JSON string. The specific format is negotiated by the NA and device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.6.5 Querying Command Revocation Tasks

Typical Scenario

After delivering a command revocation command to a device, an NA can call this API to query the execution status of the command revocation task.

API Function

This API is used by an NA to query the information and status of one or more command revocation tasks based on specified conditions on the IoT platform.

API Description

```
QueryDeviceCmdCancelTaskOutDTO2
queryDeviceCmdCancelTask(QueryDeviceCmdCancelTaskInDTO2 qdcctInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcctInDTO	Mandatory	QueryDeviceCmdCancelTaskInDTO2	query	For details, see QueryDeviceCmdCancelTaskInDTO2 structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceCmdCancelTaskInDTO2

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to **0**. The default value is **0**.
pageSize	Optional	Integer(>=1 &&<=1000)	query	Indicates the number of records to be displayed on each page. The value ranges from 1 to 1000. The default value is **1000**.
taskId	Optional	String	query	Identifies a command revocation task.

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Optional	String	query	Identifies the device whose commands are to be revoked by the revocation task.
status	Optional	String	query	Indicates the status of the command revocation task.
startTime	Optional	String	query	Indicates the start time. Revocation tasks created later than the start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Revocation tasks created earlier than the end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
appId	Optional	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCmdCancelTaskOutDTO2 structure

Parameter	Type	Description
pagination	Pagination	Indicates pagination information. For details, see **Pagination structure**.
data	List<DeviceCommandCancelTaskRespV4>	Indicates the device command list. For details, see **DeviceCommandCancelTaskRespV4 structure**.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.

Parameter	Type	Description
totalSize	long	Indicates the total number of records, that is, the total number of commands queried in the command revocation task.

DeviceCommandCancelTaskRespV4 structure

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Identifies the device whose commands are to be revoked by the revocation task.
status	String	Indicates the status of the command revocation task. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<DeviceCommandRespV4>	Indicates a list of device commands to be revoked by the revocation task. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.

Parameter	Type	Description
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0-3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates the command parameter, which is a JSON string. The specific format is negotiated by the NA and device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.7 Command Delivery (Non-NB-IoT Commands)

2.3.7.1 Calling Device Services

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform does not cache commands but delivers commands directly. When a device is offline, the commands fail to be delivered. The formats of the delivered command need to be defined by the NAs and devices. The IoT platform encapsulates and transparently transmits the commands over this API.

NOTE

Currently, this API can be used to deliver commands only to gateways equipped with the IoT Agent or AgentLite to control non-directly connected devices under the gateways.

API Function

This API is used to immediately deliver commands to gateways equipped with the IoT Agent or AgentLite to control the gateways. This API applies to devices registered with the current application.

API Description

```
InvokeDeviceServiceOutDTO invokeDeviceService(String deviceId, String serviceId,
CommandDTO2 commandDTO, String appId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(1-64)	path	Identifies a device.
serviceId	Mandatory	String(1-64)	path	Uniquely identifies a service.
commandDTO	Mandatory	CommandDTO2 structure	body	For details, see **CommandDTO2 structure**.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

CommandDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
header	Mandatory	CommandNA2CloudHeader	body	For details, see **CommandNA2CloudHeader structure**.
body	Optional	Object	body	Indicates the message body. The content of the JsonObject is a list of key-value pairs. Every key is the **paraName** of a command defined in the profile.

CommandNA2CloudHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Optional	String(0-128)	body	Identifies a command. The value of this parameter must be unique.
mode	Mandatory	Enum	body	Indicates whether an ACK message is required. ● **NOACK**: No ACK message is required. ● **ACK**: An ACK message is required. ● Other values: invalid

Parameter	Mandatory or Optional	Type	Location	Description
from	Optional	String(128)	body	Indicates the address of the message sender. ● Request initiated by an application: /users/{userId} ● Request initiated by an NA: /{serviceName} ● Request initiated by the IoT platform: /cloud/{serviceName}
toType	Optional	Enum	body	Indicates the type of the message recipient. The value options are **CLOUD** and **GATEWAY**.
to	Optional	String(128)	body	Indicates the address of the message recipient.
method	Mandatory	String(1-32)	body	Indicates the command name. For example, a **DISCOVERY** command is used to discover non-directly connected devices, and a **REMOVE** command is used to delete non-directly connected devices.
callbackURL	Optional	String(1024)	body	Indicates the callback URL.

Response Parameters

InvokeDeviceServiceOutDTO structure

Parameter	Type	Description
status	String(128)	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been delivered. This value is returned when **toType** is set to **CLOUD**. ● **failed**: The command fails to be delivered. This value is returned when **toType** is set to **CLOUD**.
timestamp	String(128)	Indicates the timestamp used for sending a command. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Type	Description
requestId	String(128)	Identifies a device command. ● When **toType** is set to **GATEWAY**, if **requestId** is carried in a request, the response carries the same **requestId** as the request; if **requestId** is not carried in a request, the IoT platform allocates a sequence number for the response. ● When **toType** is set to **CLOUD**, the value of this parameter is null.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the gateway is normal.

HTTP Status Code	Error Code	Error Description	Remarks
200	100432	The device command is muted.	The device command is muted. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	102203	CommandName is invalid.	The command name is invalid. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
403	100450	The gateway is not online.	The gateway is offline. Recommended handling: Check whether the connection between the gateway and the IoT platform is normal.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100444	The serviceType is not exist.	The service type does not exist. Recommended handling: Check whether the service type carried in the API request parameter **toType** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

2.3.8 Data Collection

The IoT platform allows NAs to query basic information about a device and view historical data reported by the device by hour, day, or month.

2.3.8.1 Querying Information About a Device

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of a device that has been registered on the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information of a specified device based on the device ID on the IoT platform, such as configuration, status and service attributes.

API Description

```
QuerySingleDeviceInfoOutDTO querySingleDeviceInfo(String deviceId, String select,
String appId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
select	Optional	String	query	Indicates the query condition. The value can be **IMSI**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

QuerySingleDeviceInfoOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Identifies a device.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.
deviceInfo	DeviceInfo	Indicates information about the device. For details, see **DeviceInfo structure**.
services	List<DeviceService>	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Identifies a device.

Parameter	Type	Description
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	String(256)	Indicates the device status details. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Type	Description
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
serviceInfo	ServiceInfo	Indicates the masked device service information. For details, see **ServiceInfo structure**.
data	ObjectNode(2097152)	Indicates an attribute-value pair (AVP).

Parameter	Type	Description
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the device command list.

Error Codes

HTTP Status Code	Error Codes	Error Description	Remarks
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Codes	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.8.2 Querying a Device Information List

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of multiple devices that have been registered on the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information (such as configuration, status and service attributes) of multiple devices based on specified conditions on the IoT platform.

API Description

```
QueryBatchDevicesInfoOutDTO queryBatchDevicesInfo(QueryBatchDevicesInfoInDTO
qbdiInDTO, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qbdiInDTO	Mandatory	QueryBatchDevicesInfoInDTO	query	For details, see **PostDeviceCommandInDTO2 structure**.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

PostDeviceCommandInDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
gatewayId	Optional	String	query	Identifies a gateway.
nodeType	Optional	String	query	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
deviceType	Optional	String	query	Indicates the device type.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.

Parameter	Mandatory or Optional	Type	Location	Description
status	Optional	String	query	Indicates the device status. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline. ● **ABNORMAL**: The device is abnormal.
startTime	Optional	String	query	Indicates the start time. Records with the device registration time later than the start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Records with the device registration time earlier than the end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
sort	Optional	String	query	Indicates the sorting mode of queried records. ● **ASC**: Records are sorted in ascending order of device registration time. ● **DESC**: Records are sorted in descending order of device registration time. The default value is **DESC**.
select	Optional	String	query	Indicates the record to be returned. The value can be **IMSI**.

Response Parameters

QueryBatchDevicesInfoOutDTO structure

Parameter	Type	Description
totalCount	long	Indicates the number of queried records.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records on each page.
devices	List< QuerySingleDeviceInfoOutDTO>	Indicates the device pagination information list. For details, see **QuerySingleDeviceInfoOutDTO structure**.

QuerySingleDeviceInfoOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Identifies a device.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.
deviceInfo	DeviceInfoQueryDTO	Indicates information about the device. For details, see **DeviceInfo structure**.
services	List<DeviceService>	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Type	Description
model	String(256)	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, and **ABNORMAL**.
statusDetail	String(256)	Indicates the device status details. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail structure**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.

Parameter	Type	Description
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
serviceInfo	ServiceInfo	Indicates the masked device service information. For details, see **ServiceInfo structure**.
data	ObjectNode(2097152)	Indicates an attribute value pair.
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the device command list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100218	The gatewayId and pageNo cannot be both null.	The **gatewayId** and **pageNo** parameters cannot be null at the same time. Recommended handling: Check whether **gatewayId** or **pageNo** is set.
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.8.3 Historical Device Data Query

Typical Scenario

The IoT platform receives and saves service data reported by devices during daily operation. The storage duration of device data can be configured by calling the API for modifying device information and the device data can be stored for a maximum of 90 days. If an NA needs to view the historical data reported by a device to the IoT platform, the NA can call this API to obtain the data.

API Function

This API is used by an NA to query historical data reported by a specified device to the IoT platform based on the device ID.

API Description

```
QueryDeviceDataHistoryOutDTO queryDeviceDataHistory(QueryDeviceDataHistoryInDTO
qddhInDTO, String accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qddhInDTO	Mandatory	QueryDeviceDataHistoryInDTO structure	query	For details, see QueryDeviceData-HistoryInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceDataHistoryInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Mandatory	String	query	Identifies a device.
gatewayId	Mandatory	String	query	Identifies a gateway.
serviceId	Optional	String	query	Identifies a service.
property	Optional	String	query	Indicates the service attribute.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.

Parameter	Mandatory or Optional	Type	Location	Description
startTime	Optional	String	query	Indicates the start time. Historical shadow data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Historical shadow data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDataHistoryOutDTO

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
deviceDataHistoryDTOs	List<DeviceDataHistoryDTO>	Indicates the historical device data list. For details, see **DeviceDataHistoryDTO structure**.

DeviceDataHistoryDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Identifies a device.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
appId	String(256)	Uniquely identifies an NA.
data	JsonObject	Indicates the data reported by the device.

Parameter	Type	Description
timestamp	String(256)	Indicates the timestamp when the data is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description. For example, check whether the value of **pageSize** exceeds 2000.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.8.4 Querying Historical Device Shadow Data

Typical Scenario

When an NA modifies the configuration of a device shadow by calling the API for modifying device shadow information, the IoT platform saves the modification record. If the NA needs to view historical configuration records of the device shadow, the NA can call this API to obtain the records.

API Function

This API is used by an NA to query historical configuration data about a device shadow based on the device ID.

API Description

```
QueryDeviceDesiredHistoryOutDTO
queryDeviceDesiredHistory(QueryDeviceDesiredHistoryInDTO qddhInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qddhInDTO	Mandatory	QueryDeviceDesiredHistoryInDTO structure	query	For details, see QueryDeviceDesiredHistoryInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceDesiredHistoryInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Mandatory	String	query	Identifies a device.
gatewayId	Mandatory	String	query	Identifies a gateway.
serviceId	Optional	String	query	Identifies a service.
property	Optional	String	query	Indicates the service attribute.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.

Parameter	Mandatory or Optional	Type	Location	Description
startTime	Optional	String	query	Indicates the start time. Historical shadow data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Historical shadow data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDesiredHistoryOutDTO structure

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
DeviceDesiredHistoryDTO	List<DeviceDesiredHistoryDTO>	Indicates the historical configuration data about a device shadow. For details, see **DeviceDesiredHistoryDTO structure**.

DeviceDesiredHistoryDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Identifies a device.
gatewayId	String(256)	Identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is a non-directly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates.
appId	String(256)	Uniquely identifies an NA.
desired	JsonObject	Indicates the configuration information delivered to the device.

Parameter	Type	Description
timestamp	String(256)	Indicates the timestamp when the data is configured. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.8.5 Querying Service Capabilities of a Device

Typical Scenario

If an NA needs to know which service attributes can be reported by a device and which commands can be delivered to the device, the NA can call this API to query the device service capabilities defined in the profile file of the device on the IoT platform.

API Function

This API is used by an NA to query device service capabilities, such as service attributes and device commands.

API Description

```
QueryDeviceCapabilitiesOutDTO
queryDeviceCapabilities(QueryDeviceCapabilitiesInDTO qdcInDTO, String
accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcInDTO	Mandatory	QueryDeviceCapabilitiesInDTO structure	query	For details, see QueryDeviceCapabilitiesInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceCapabilitiesInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Optional	String	query	Identifies a gateway.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Optional	String	query	Identifies a device.

Response Parameters

QueryDeviceCapabilitiesOutDTO structure

Parameter	Type	Description
deviceCapabilities	List<DeviceCapabilityDTO>	Indicates the query result list. For details, see **DeviceCapabilityDTO structure**.

DeviceCapabilityDTO structure

Parameter	Type	Description
deviceId	String(256)	Identifies a device.
serviceCapabilities	List<ServiceCapabilityDTO>	Indicates the service capability list. For details, see **ServiceCapabilityDTO structure**.

ServiceCapabilityDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
option	String(256)	Indicates a service option.
description	String(10240)	Indicates the service description.
commands	List<ServiceCommand>	Indicates the supported commands. For details, see **ServiceCommand structure**.
properties	List<ServiceProperty>	Indicates the attribute list. For details, see **ServiceProperty structure**.

ServiceCommand structure

Parameter	Type	Description
commandName	String(256)	Indicates the command name.
paras	List<ServiceCommandPara>	Indicates the attribute list. For details, see **ServiceCommandPara structure**.
responses	List<ServiceCommandResponse>	Indicates the response list. For details, see **ServiceCommandResponse structure**.

ServiceCommandPara structure

Parameter	Type	Description
paraName	String(256)	Indicates the parameter name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.
min	String	Indicates the minimum value of the attribute.

Parameter	Type	Description
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

ServiceCommandResponse structure

Parameter	Type	Description
responseName	String(256)	Indicates the response name.
paras	List<ServiceCommandPara>	Indicates the attribute list. For details, see **ServiceCommandPara structure**.

ServiceProperty structure

Parameter	Type	Description
propertyName	String(256)	Indicates the attribute name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.
min	String	Indicates the minimum value of the attribute.
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
method	String(256)	Indicates the access method. The values are as follows: ● **R**: Readable ● **W**: Writable ● **E**: Observable
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9 Device Group Management

2.3.9.1 Creating a Device Group

Typical Scenario

An NA can call this API to create device groups on the IoT platform, and allocate devices to different device groups for group management. A device can be bound to multiple device groups.

When the NA needs to perform operations on devices (such as upgrading device software and firmware or delivering commands to devices in batches), the NA can select devices to be operated by device group.

API Function

This API is used by an NA to create device groups on the IoT platform to manage devices by group.

API Description

```
CreateDeviceGroupOutDTO createDeviceGroup(CreateDeviceGroupInDTO cdgInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cdgInDTO	Mandatory	CreateDeviceGroupInDTO structure	body	For details, see **CreateDeviceGroupInDTO structure**.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

CreateDeviceGroupInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.
appId	Optional	String (50)	body	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
maxDevNum	Optional	Integer (>=0)	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.
deviceIds	Optional	List<String>	body	Identifies the devices to be added to the device group.
id	Optional	String(1-50)	Body	Identifies a device group.

Response Parameters

CreateDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group. The device group ID is automatically generated by the IoT platform.
appId	String(50)	Uniquely identifies an NA.
maxDevNum	Integer (>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100607	The devGroup has reached the limit.	The number of device groups reaches the limit. Recommended handling: Check whether the number of created device groups reaches the upper limit specified in the license.
400	100609	Too much devices to add.	Too many devices are added to the device group. Recommended handling: Ensure that the number of device IDs contained in **deviceIds** is within the range specified by **maxDevNum**.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.2 Deleting a Device Group

Typical Scenario

If a device group is no longer needed on the IoT platform due to group changes, an NA can call this API to delete a specified device group.

API Function

This API is used by an NA to delete the configuration information about a device group by device group ID on the IoT platform.

API Description

```
void deleteDeviceGroup(String devGroupId, String accessAppId, String accessToken)
throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed	The device group does not exist. Recommended handling: Check whether the device group ID is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.3 Modifying a Device Group

Typical Scenario

If information about a device group (such as the device group name and the device quantity limit in the device group) needs to be modified due to service changes, an NA can call this API to modify the information.

API Function

This API is used to modify the information of a specified device group on the IoT platform.

API Description

```
ModifyDeviceGroupOutDTO modifyDeviceGroup(ModifyDeviceGroupInDTO mdgInDTO, String
devGroupId, String accessAppId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
mdgInDTO	Mandatory	ModifyDeviceGroupInDTO structure	body	For details, see **ModifyDeviceGroupInDTO**.

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String (50)	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Mandatory	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

ModifyDeviceGroupInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.
maxDevNum	Optional	Integer(>=0)	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.

Response Parameters

ModifyDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies the device group.
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.

HTTP Status Code	Error Code	Error Description	Remarks
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.4 Querying Details About a Device Group

Typical Scenario

An NA can call this API to query information about all the created device groups to check the group details and usage of the device groups.

API Function

This API is used by an NA to query information about all created device groups on the IoT platform.

API Description

```
QueryDeviceGroupsOutDTO queryDeviceGroups(QueryDeviceGroupsInDTO qdgInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdgInDTO	Mandatory	QueryDeviceGroupsInDTO	query	For details, see **QueryDeviceGroupsInDTO**.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceGroupsInDTO

Parameter	Mandatory or Optional	Type	Location	Description
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of device group records on each page. The default value is **1**.
name	Optional	String	query	Indicates the device group name.

Response Parameters

QueryDeviceGroupsOutDTO structure

Parameter	Type	Description
totalCount	long	Indicates the total number of device groups.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of device group records on each page.
list	List<QuerySingleDeviceGroupOutDTO>	Indicates details about the device group. For details, see **QuerySingleDeviceGroupOutDTO structure**.

QuerySingleDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies the device group.
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.
creator	String(1-50)	Indicates the name of the user who created the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.5 Querying Information About a Specified Device Group

Typical Scenario

An NA can call this API to query information about a specified device group to check the usage of the device group.

API Function

This API is used by an NA to query the information about a device group by device group ID on the IoT platform.

API Description

```
QuerySingleDeviceGroupOutDTO querySingleDeviceGroup(String devGroupId, String
accessAppId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

QuerySingleDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies the device group.
appId	String (50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group.
curDevNum	Integer	Indicates the current number of devices in the device group.

Parameter	Type	Description
creator	String(1-50)	Indicates the name of the user who created the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.6 Querying Members in a Specified Device Group

Typical Scenario

An NA can call this API to query information about members in a specified device group.

API Function

This API is used by an NA to query the information about a device in a specified device group on the IoT platform.

API Description

```
QueryDeviceGroupMembersOutDTO
queryDeviceGroupMembers(QueryDeviceGroupMembersInDTO qdgmInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdgmInDTO	Mandatory	QueryDeviceGroupMembersInDTO structure	query	For details, see **QueryDeviceGroupMembersInDTO structure**.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryDeviceGroupMembersInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	query	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.

Parameter	Mandatory or Optional	Type	Location	Description
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer(1000)	query	Indicates the number of devices on each page. The default value is **10**.

Response Parameters

QueryDeviceGroupMembersOutDTO structure

Parameter	Type	Description
totalCount	long	Indicates the number of devices in the device group.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of devices on each page.
deviceIds	List<String>	Identifies the devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	107001	The serviceId is not exist.	The service ID does not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.7 Adding Members to a Device Group

Typical Scenario

An NA can call this API to add a new device or an existing device to a specified device group. Before adding a device to a device group, you are advised to query the current number of devices and the maximum number of devices allowed in the device group by calling the API for querying information about a specified device group.

API Function

This API is used by an NA to add devices to a specified device group on the IoT platform.

API Description

```
DeviceGroupWithDeviceListDTO addDevicesToGroup(DeviceGroupWithDeviceListDTO
dgwdlDTO, String accessAppId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dgwdlDTO	Mandatory	DeviceGroupWithDeviceListDTO structure	body	For details, see DeviceGroupWithDeviceListDTO structure.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

DeviceGroupWithDeviceListDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies the devices to be added to the device group.

Response Parameters

DeviceGroupWithDeviceListDTO structure

Parameter	Type	Description
devGroupId	String(1-50)	Identifies a device group.
deviceIds	List<String>	Identifies the devices to be added to the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.9.8 Deleting Members from a Device Group

Typical Scenario

If one or more devices in a device group do not belong to the device group any longer, an NA can call this API to delete them from the device group.

API Function

This API is used by an NA to delete devices from a specified device group on the IoT platform.

API Description

```
void deleteDevicesFromGroup(DeviceGroupWithDeviceListDTO dgwdlDTO, String
accessAppId, String accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dgwdlDTO	Mandatory	DeviceGroupWithDeviceListDTO structure	body	For details, see DeviceGroupWithDeviceListDTO structure.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

DeviceGroupWithDeviceListDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies devices to be deleted from the device group.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

2.3.10 Device Upgrade

2.3.10.1 Querying a Version Package List

Typical Scenario

Before upgrading the device version, an NA can call this API to query the version upgrade packages that have been uploaded to the IoT platform to ensure that the target version package has been uploaded.

API Function

This API is used by an NA to query a list of uploaded version packages that meet a specified condition.

API Description

```
QueryUpgradePackageListOutDTO
queryUpgradePackageList(QueryUpgradePackageListInDTO quplInDTO, String
accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
quplInDTO	Mandatory	QueryUpgradePackageListInDTO structure	query	For details, see QueryUpgradePackageListInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryUpgradePackageListInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileType	Optional	String(256)	query	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	Optional	String(256)	query	Indicates the type of the device to which the version package is applicable.
model	Optional	String(256)	query	Indicates the model of the device to which the version package is applicable.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device to which the version package is applicable.
version	Optional	String(256)	query	Indicates the version of the version package.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradePackageListOutDTO structure

Parameter	Type	Description
data	List<QueryUpgradePackageOutDTO>	Indicates the version package list. For details, see **QueryUpgradePackageOutDTO structure**.
pageNo	Integer	Indicates the page number.
pageSize	Integer	Indicates the number of records to be displayed on each page.
totalCount	Integer	Indicates the total number of query results.

QueryUpgradePackageOutDTO structure

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.

Parameter	Type	Description
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.
description	String	Indicates the version package description.
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the values of **pageNo** and **pageSize** in the API request are within the valid ranges.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

2.3.10.2 Querying a Specified Version Package

Typical Scenario

Before upgrading the device version, an NA can call this API to query the target version upgrade package to ensure that it has been uploaded to the IoT platform.

API Function

This API is used by an NA to query a specified version package based on the version package ID on the IoT platform. The version package ID can be obtained by calling the API for querying a version package list.

API Description

```
QueryUpgradePackageOutDTO queryUpgradePackage(String fileId, String accessToken)
throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

QueryUpgradePackageOutDTO structure

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.

Parameter	Type	Description
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.
description	String	Indicates the version package description.
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

2.3.10.3 Deleting a Specified Version Package

Typical Scenario

If a device version package is no longer needed, an NA can call this API to delete the version package from the IoT platform.

API Function

This API is used by an NA to delete a specified version package from the IoT platform based on the version package ID. The ID of the version package to be deleted can be obtained by calling the API for querying a version package list.

API Description

```
void deleteUpgradePackage(String fileId, String accessToken) throws
NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

2.3.10.4 Creating a Software Upgrade Task

Typical Scenario

If the device software needs to be upgraded, an NA can call this API to create a software upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the software of multiple devices on the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Description

```
CreateUpgradeTaskOutDTO createSoftwareUpgradeTask(CreateUpgradeTaskInDTO
cutInDTO, String accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cutInDTO	Mandatory	CreateUpgradeTaskInDTO structure	body	For details, see CreateUpgradeTaskInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

CreateUpgradeTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	body	Identifies the target software version.
targets	Mandatory	OperateDevices	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.
policy	Optional	OperatePolicy	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.

Parameter	Mandatory or Optional	Type	Location	Description
deviceType	Optional	String	body	Indicates the device type. This parameter must be specified when a device group is specified.
model	Optional	String	body	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	Optional	String	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies an operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.

HTTP Status Code	Error Code	Error Description	Remarks
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

2.3.10.5 Creating a Firmware Upgrade Task

Typical Scenario

If the device firmware needs to be upgraded, an NA can call this API to create a firmware upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the firmware of multiple devices on the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Description

```
CreateUpgradeTaskOutDTO createFirmwareUpgradeTask(CreateUpgradeTaskInDTO
cutInDTO, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cutInDTO	Mandatory	CreateUpgradeTaskInDTO structure	body	For details, see CreateUpgradeTaskInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

CreateUpgradeTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	body	Identifies the target firmware version.
targets	Mandatory	OperateDevices	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.
policy	Optional	OperatePolicy	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.

Parameter	Mandatory or Optional	Type	Location	Description
deviceType	Optional	String	body	Indicates the device type. This parameter must be specified when a device group is specified.
model	Optional	String	body	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	Optional	String	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies the operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.

HTTP Status Code	Error Code	Error Description	Remarks
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

2.3.10.6 Querying the Result of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, an NA can call this API to query details about the upgrade task, including the configuration and execution status.

API Function

This API is used by an NA to query details about a software or firmware upgrade task, including the configuration and execution status.

API Description

```
QueryUpgradeTaskOutDTO queryUpgradeTask(String operationId, String accessToken)
throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

Response Parameters

QueryUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies the operation task.
createTime	String	Indicates the time when the operation task was created.
startTime	String	Indicates the time when the operation task was started.
stopTime	String	Indicates the time when the operation task was stopped.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
targets	OperateDevices	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
policy	OperatePolicy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.

Parameter	Type	Description
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	OperationStaResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
extendPara	JsonString	Indicates an extended parameter of the operation task. For details, see **extendPara request parameter**.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Indicates the device type. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The operation task is executed now. ● **device_online**: The operation task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.

Parameter	Type	Description
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

extendPara request parameter when **operateType** is set to **softwareUpgrade** or **firmwareUpgrade**

Parameter	Type	Description
fileVersion	String	Identifies the target version.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

2.3.10.7 Querying Details About Subtasks of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, the upgrade of each device involved in the task is a subtask (the number of subtasks is the same as that of the devices

involved in the task). An NA can call this API to query details about subtasks of the upgrade task to check their execution status.

API Function

This API is used by an NA to query upgrade status of each device involved in a software or firmware upgrade task.

API Description

```
QueryUpgradeSubTaskOutDTO queryUpgradeSubTask(QueryUpgradeSubTaskInDTO qustInDTO,
String operationId, String accessToken) throws NorthApiException
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qustInDTO	Mandatory	QueryUpgradeSubTaskInDTO structure	query	For details, see QueryUpgradeSubTaskInDTO structure.
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryUpgradeSubTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
subOperationStatus	Optional	String	query	Indicates subtask status. If this parameter is not specified, execution details of all subtasks under the operation task are queried. ● **wait**: The subtask is waiting to be executed. ● **processing**: The subtask is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The subtask is successfully executed. ● **stop**: The subtask stops being executed.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeSubTaskOutDTO structure

Parameter	Type	Description
data	list<SubOperationInfo>	Indicates the subtask list. For details, see **SubOperationInfo structure**.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalCount	long	Indicates the total number of query results.

SubOperationInfo structure

Parameter	Type	Description
subOperationId	String	Identifies a subtask.
createTime	String	Indicates the time when the subtask was created.
startTime	String	Indicates the time when the subtask was started.
stopTime	String	Indicates the time when the subtask was stopped.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
deviceId	String	Identifies a device on which the subtask is performed.
status	String	Indicates the subtask status. ● **wait**: The subtask is waiting to be executed. ● **processing**: The subtask is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The subtask is successfully executed. ● **stop**: The subtask stops being executed.
detailInfo	String	Indicates detailed description of the subtask status. In case of a failure, the value of this parameter is the failure cause.
extendInfo	JsonString	Indicates extended information of the subtask. The value of this parameter varies depending on the operation type.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

2.3.10.8 Querying an Upgrade Task List

Typical Scenario

An NA can call this API to query the created upgrade tasks to view the detailed information and execution status of each upgrade task.

API Function

This API is used by an NA to query details about upgrade tasks that meet specified conditions.

API Description

```
QueryUpgradeTaskListOutDTO queryUpgradeTaskList(QueryUpgradeTaskListInDTO
qutlInDTO, String accessToken) throws NorthApiException
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qutlInDTO	Mandatory	QueryUpgradeTaskListInDTO structure	query	For details, see QueryUpgradeTaskListInDTO structure.
accessToken	Mandatory	String	header	If the **Periodically Refreshing a Token** API is called, set this parameter to **null**. Otherwise, set this parameter to the accessToken obtained by the **Authentication** API.

QueryUpgradeTaskListInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
operationType	Optional	String(256)	query	Indicates the operation type. ● firmware_upgrade ● software_upgrade
operationStatus	Optional	String(256)	query	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
deviceType	Optional	String(256)	query	Indicates the device type for which the operation task is intended.
model	Optional	String(256)	query	Indicates the device model for which the operation task is intended.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device for which the operation task is intended.

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Optional	String(256)	query	Identifies the device for which the operation task is intended.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeTaskListOutDTO structure

Parameter	Type	Description
data	List<OperationInfo>	Indicates the task list. For details, see **OperationInfo structure**.
pageNo	Integer	Indicates the page number.
pageSize	Integer	Indicates the number of records to be displayed on each page.
totalCount	Integer	Indicates the total number of query results.

OperationInfo structure

Parameter	Type	Description
operationId	String	Identifies an operation task.
createTime	String	Indicates the time when the operation task was created.
startTime	String	Indicates the time when the operation task was started.

Parameter	Type	Description
stopTime	String	Indicates the time when the operation task was stopped.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
targets	OperateDevices	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
policy	OperatePolicy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	OperationStaResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
extendPara	JsonString	Indicates extended information of the operation task. The value of this parameter varies depending on the operation type.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Indicates the device type. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Type	Description
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The operation task is executed now. ● **device_online**: The operation task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

Error Code

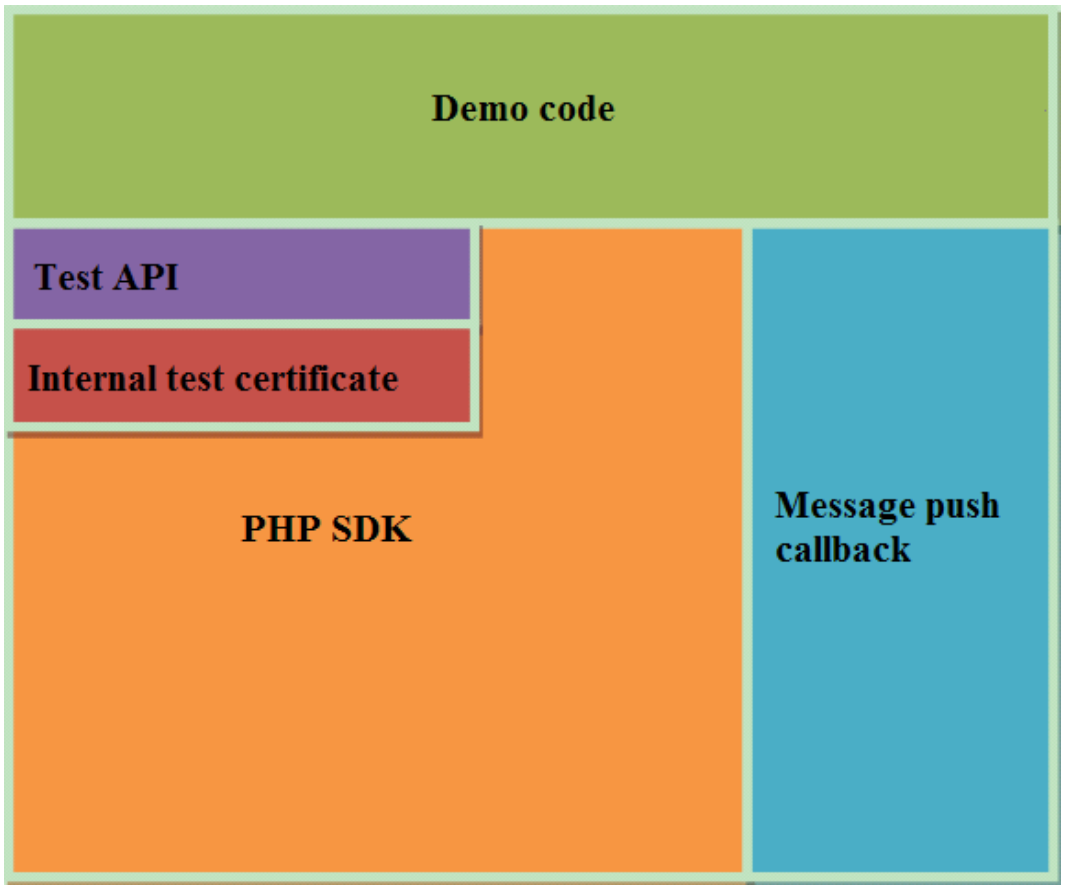
HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

3 Northbound PHP SDK API Reference

3.1 SDK Demo Architecture and Usage Guide

- The demo code is the sample code used for calling SDK APIs, including initializing and calling each API. The demo code is for your reference only.
- The PHP SDK uses PHP methods to call northbound RESTful APIs provided by the IoT platform to communicate with the platform.
- The message push callback uses PHP code to implement the callback API, which is called by the IoT platform to push messages to application servers. An application inherits the PushMessageReceiver class and rewrites the methods in the class to receive the content of a push message.
- The test API tests whether the basic functions between the SDK and IoT platform are available and generates test results. During the test, the internal test certificate and the certificate set by developers are used.



3.2 SDK Initialization Configuration and Test

3.2.1 Methods of the NorthApiClient Class

The methods of the NorthApiClient class are used to create an application instance. They are the prerequisites for calling other SDK APIs. The following describes the main methods:

Method	Description
public function __set($name, $value)	Initializes the attributes in the NorthApiClient class. For example, $northClient->$clientInfo($clientInfo).
public function initSSLConfig0()	Initializes the two-way authentication configuration. Other methods can only be used after this method is called. **NOTICE** This method uses the test certificate, which is not the formal certificate and does not verify the host name. Therefore, this method is used only in the testing phase of integration interconnection.

Method	Description
public function initSSLConfig1($sslConfig)	Initializes the two-way authentication configuration. Other methods can only be used after this method is called. For details about the SSLConfig class, see **2.4 Methods of the SSLConfig Class**. **NOTICE** This method is used to import the certificate and can be applied for at the commercial phase or official phase.
public function getVersion()	Queries the SDK version.

3.2.2 Methods of the ClientInfo Class

The methods of the ClientInfo class are used to set the basic information about the interconnection. The following describes the main methods (excluding the get method):

Method	Description
public function __set($name, $value)	Sets the IP address and port number of the IoT platform, application ID, and secret for interconnection. For example, $clientInfo->appId = 'zxcxxxxxxx12'.

3.2.3 Methods of the NorthApiException Class

An exception is reported when an error occurs during SDK processing or platform request processing. The exception is encapsulated into an NorthApiException object and then reported to the method invoker. The following describes the main methods (excluding the set method):

Method	Description
public function __get($name)	Obtains the attribute information of the exception, for example, the error code ($e->error_code).

3.2.4 Methods of the SSLConfig Class

The methods of the SSLConfig class are used to set the certificate path and password. The following describes the main methods (excluding the get method):

Method	Description
public function __set($name, $value)	Sets the absolute path of a client certificate, for example, $sslConfig-> selfCertPath = './client.pem'.

3.3 Service API List

3.3.1 Secure Application Access

After an NA obtains authentication information and connects to the IoT platform, the NA uses the authentication information to call other APIs.

3.3.1.1 Authentication

Typical Scenario

This API is called by an NA for access authentication when the NA accesses open APIs of the IoT platform for the first time. After the authentication of the NA expires, the NA must call this API to perform authentication again so that the NA can continue to access open APIs of the IoT platform.

API Function

This API is used by an NA to get authenticated before accessing open APIs of the IoT platform for the first time.

Note

The authentication API is the prerequisite for calling other APIs. Except the authentication API itself, other APIs must use the accessToken obtained by the authentication API.

If the access token is obtained for multiple times, the previous access token is invalid and the last access token obtained is valid. Do not obtain the access token through concurrent attempts.

API Description

```
public function getAuthToken()
```

Class

Authentication

Parameter Description

The values of **appId** and **secret** are those of the members of **Methods of the NorthApiClient Class** when the methods in **Methods of the ClientInfo Class** are called.

Return Value

AuthOutDTO

Parameter	Description
$scope	Indicates the application permission scope, that is, the scope of IoT platform resources that can be accessed using the **accessToken**. This parameter has a fixed value of **default**.
$tokenType	Indicates the type of the **accessToken**. This parameter has a fixed value of **bearer**.
$expiresIn	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
$accessToken	Indicates the authentication parameter that is used to access APIs of the IoT platform.
$refreshToken	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.1.2 Refreshing a Token

Typical Scenario

The access token obtained by calling the Authentication API has a valid time. When the access token is about to expire, an NA can call this API to obtain a new access token.

API Function

This API is used by an NA to obtain a new access token from the IoT platform when the access token is about to expire.

API Description

```
public function refreshAuthToken(AuthRefreshInDTO $arInDTO)
```

Class

Authentication

Parameter Description

AuthRefreshInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
$secret	Mandatory	body	Indicates the password that corresponds to **appId**. It is used to log in to the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
$refreshToken	Mandatory	body	Indicates the refresh token used for obtaining a new **accessToken**. The value of this parameter is obtained when the **Authentication** API is called.

Return Value

AuthRefreshOutDTO

Parameter	Description
$scope	Indicates the applied permission range. This parameter has a fixed value of **default**.
$tokenType	Indicates the access token type. This parameter has a fixed value of **bearer**.
$expiresIn	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
$accessToken	Indicates the authentication parameter that is used to access APIs of the IoT platform.
$refreshToken	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2 Device Management

An NA adds a device to the IoT platform and obtains the device ID and verification code. After connecting to the IoT platform, the device establishes a subordinate relationship with the NA.

3.3.2.1 Registering a Device (Verification Code Mode)

Typical Scenario

Before a device accesses the IoT platform by using verification code, an NA needs to call this API to register the device with the IoT platform and set a unique identification code of the device (such as the IMEI) as the verification code. Then, the device can use the unique identification code to get authenticated and connect to the IoT platform.

API Function

This API is used by an NA to register a device with the IoT platform. After registration, the device can connect to the IoT platform.

API Description

```
public function regDirectDevice($rddInDto, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$rddInDto	Mandatory	body	For details, see **RegDirectDeviceInDTO structure**.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

RegDirectDeviceInDTO

Parameter	Mandatory or Optional	Location	Description
deviceInfo	Optional	Body	Indicates the device information. For details, see **DeviceInfo structure**.
endUserId	Optional	Body	This parameter identifies an end user. In the NB-IoT solution, this parameter is set to the IMSI of the device. In the Smart Home solution, this parameter is set to the application account.
imsi	Optional	Body	Indicates the IMSI of an NB-IoT device.
isSecure	Optional	Body	Indicates whether the device is secure. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure. **NOTE** If a user needs to register a secure device, this parameter must be specified.
nodeId	Mandatory	Body	Uniquely identifies a device. The value of this parameter must be the same as the device ID reported by the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
psk	Optional	Body	If the pre-shared key (PSK) is specified in the request, the IoT platform uses the specified PSK. If the PSK is not specified in the request, the PSK is generated by the IoT platform. The value is a string of characters, including upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.

Parameter	Mandatory or Optional	Location	Description
timeout	Optional	Body	Indicates the validity period for device registration. When this API is called to register a device, the device can be bound within the validity period. If the device is not bound within the validity period, the registration information will be deleted. The value ranges from 0 to 2147483647. If this parameter is set to **0**, the device verification code is always valid. (The recommended value is **0**.) The default value is **180**. The default value can be configured. For details, contact the IoT platform maintenance personnel. Unit: second
verifyCode	Conditionally optional Mandatory	body	Indicates the device verification code. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is generated by the IoT platform. In the NB-IoT solution, this parameter is mandatory and must be set to the same value as **nodeId**.
productId	Optional	Body	Identifies the product to which the device belongs.

DeviceInfo structure

Parameter	Mandatory or Optional	Location	Description
manufacturerId	Optional	Body	Uniquely identifies a manufacturer.
manufacturerName	Optional	Body	Indicates the manufacturer name.
deviceType	Optional	Body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Mandatory or Optional	Location	Description
model	Mandatory	Body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
protocolType	Optional	Body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Return Value

RegDirectDeviceOutDTO

Parameter	Description
$deviceId	Identifies a device.
$verifyCode	Indicates a verification code that can be used by the device to obtain the device ID and password. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
$timeout	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.
$psk	Indicates a random PSK. If the PSK is carried in the request, it is used. Otherwise, the IoT platform generates a random PSK.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	103028	The license pool resources.	The license resources have been used up.

HTTP Status Code	Error Code	Error Description	Remarks
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100416	The device has already been bounded.	The device has been bound to the IoT platform. Recommended handling: Check whether the device is registered.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application has not been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	600002	The product not existed.	The product does not exist. Recommended handling: Check whether **productId** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100412	The amount of device has reached the limit.	The number of devices under the current application reaches the upper limit. Recommended handling: Check whether the number of devices under the current application reaches the upper limit.

HTTP Status Code	Error Code	Error Description	Remarks
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	103026	The license is not exist.	The license does not exist. Recommended handling: An internal license error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.2 Refreshing a Device Key

Typical Scenario

If the unique identification code of a device that has been registered with the IoT platform changes (for example, a device is replaced), an NA needs to call this API to update the unique identification code of the device and rebind the device.

NOTE

The device password can be updated only when the device is offline.

API Function

This API is used by an NA to update the node ID of a device that has been registered with the IoT platform, and rebind the device with the device ID unchanged.

API Description

```
public function refreshDeviceKey($rdkInDTO, $deviceId, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$rdkInDTO	Mandatory	body	For details, see **RegDirectDeviceInDTO structure**.
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

RefreshDeviceKeyInDTO structure

Parameter	Mandatory or Optional	Location	Description
$verifyCode	Optional	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. You are advised to set this parameter to the value of **nodeId**.

Parameter	Mandatory or Optional	Location	Description
$nodeId	Optional	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. ● If the value is **null**, the value of this parameter remains unchanged. ● If the value is not **null**, the value of this parameter is updated. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
$timeout	Optional	body	Indicates the validity period of the verification code, in units of seconds. The value is an integer greater than or equal to **0**. ● If this parameter is set to **null**, the default value **180** prevails. ● If this parameter is set to **0**, the verification code never expires. ● If this parameter is not set to **0**, the verification code expires after the specified time elapses.

Return Value

RefreshDeviceKeyOutDTO structure

Parameter	Description
$verifyCode	Indicates a verification code that can be used by the device to obtain the device ID and password. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
$timeout	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	100610	Device is not active.	The device has not been activated. Recommended handling: Check whether the device has been connected to the IoT platform and activated.
400	100611	Device is online.	The device is online. Recommended handling: Enable the device to go offline or disconnect the device from the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.3 Modifying Device Information

Typical Scenario

After an NA registers a device with the IoT platform and the basic information about the device changes, the NA can also call this API to modify device information on the IoT platform.

API Function

This API is used to modify the basic information about a device, including the device type, device model, manufacturer, and access protocol.

API Description

```
public function  modifyDeviceInfo($mdiInDto, $deviceId, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$mdiInDto	Mandatory	body	For details, see **ModifyDeviceInforInDTO structure**.
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceInforInDTO structure

Parameter	Mandatory or Optional	Location	Description
$customFields	Optional	body	User-defined field list. Users can set customized fields. For details, see **CustomField structure**.
$deviceConfig	Optional	body	Indicates device configuration information. For details, see **DeviceConfigDTO structure**.
$deviceType	Optional	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**. In the NB-IoT solution, the device type must be changed to be the same as that defined in the profile after a device is registered.
$endUser	Optional	body	Indicates an end user. If the device is a directly connected device, this parameter is optional. If it is a non-directly connected device, this parameter can be set to **null**.
$location	Optional	body	Indicates the device location.

Parameter	Mandatory or Optional	Location	Description
$manufacturerId	Optional	body	Uniquely identifies a manufacturer. In the NB-IoT solution, the manufacturer ID must be changed to be the same as that defined in the profile after a device is registered.
$manufacturerName	Optional	body	Indicates the manufacturer name.
$model	Optional	body	Indicates the device model. In the NB-IoT solution, the model must be changed to be the same as that defined in the profile after a device is registered.
$mute	Optional	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
$name	Optional	body	Indicates the device name.
$organization	Optional	body	Indicates the organization to which the device belongs.
$protocolType	Optional	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
$region	Optional	body	Indicates the region information about a device.
$timezone	Optional	body	Indicates the time zone where the device is located. The time zone code is used. For example, the time zone code of Beijing time zone is Asia/Beijing.
$imsi	Optional	body	Indicates the IMSI of an NB-IoT device.
$ip	Optional	body	Indicates the device IP address.

Parameter	Mandatory or Optional	Location	Description
$isSecure	Optional	body	Indicates the security status of a specified device. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure.
$psk	Optional	body	Indicates the PSK. The value is a string of characters that consist of only upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.
$tags	Optional	body	Indicates the device tag information. For details, see **Tag2 structure**.

CustomField structure

Parameter	Mandatory or Optional	Location	Description
fieldName	Optional	Body	Indicates the field name.
fieldType	Optional	Body	Indicates the field type.
fieldValue	Optional	Body	Indicates the field value.

DeviceConfigDTO structure

Parameter	Mandatory or Optional	Description
$dataConfig	Optional	Indicates data configuration information. For details, see **DataConfigDTO structure**.

DataConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
$dataAgingTime	Optional	Integer	Indicates the data aging time. The value range is 0-90. Unit: day.

Tag2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.
tagType	Optional	Integer	body	Indicates the tag type.

Return Value

```
response:
Status Code: 200 OK
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100440	The isSecure is invalid.	The value of **isSecure** is incorrect.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	500004	The amount of frozen devices has reached the limit.	The number of frozen devices has reached the upper limit.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.4 Deleting a Device (Verification Code Mode)

Typical Scenario

If a device that has been registered with the IoT platform does not need to connect to the platform, an NA can call this API to delete the device. If the device needs to connect to the IoT platform again, the NA must register the device again.

API Function

This API is used by an NA to delete a registered device from the IoT platform.

API Description

```
public function deleteDirectDevice($deviceId, $cascade, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.

Parameter	Mandatory or Optional	Location	Description
$cascade	Mandatory	query	This parameter is valid only when the device is connected to a non-directly connected device. If this parameter is not set, the value **null** is used. ● **true**: The directly connected device and the non-directly-connected devices connected to it are deleted. ● **false**: The directly connected device is deleted but the non-directly-connected devices connected to it are not deleted. In addition, the attribute of the non-directly-connected devices is changed to directly-connected.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

```
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.5 Querying Device Activation Status

Typical Scenario

After an NA registers a device on the IoT platform, the activation status of the device is **false** before the device connects to the IoT platform for the first time. When the device connects to the IoT platform for the first time, the activation status of the device is **true** regardless of whether the device is online, offline, or abnormal. The NA can call this API to query the activation status of the device to check whether the device has connected to the IoT platform.

API Function

This API is used by an NA to query the activation status of a device on the IoT platform to determine whether the device has connected to the IoT platform.

API Description

```
public function queryDeviceStatus($deviceId, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

QueryDeviceStatusOutDTO

Parameter	Description
$deviceId	Identifies a device.

Parameter	Description
$activated	Indicates whether the device is activated through a verification code. ● **true**: The device is activated. ● **false**: The device is not activated.
$name	Indicates the device name.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.6 Querying Device Shadow Information

Typical Scenario

When a device is in the offline or abnormal state, an NA cannot deliver configuration information to the device by sending a command. In this case, the NA can deliver the configuration information to the device shadow. When the device goes online, the device shadow will deliver the configuration information to the device. The NA can call this API to check the device configuration information and the latest data reported by the device on the device shadow.

API Function

This API is used by an NA to query the device shadow information of a device, including the device configuration information (in the desired section) and the latest data reported by the device (in the reported section).

API Description

```
public function queryDeviceShadow($deviceId, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

Status Code: 200 OK

QueryDeviceShadowOutDTO

Parameter	Description
$deviceId	Identifies a device.
$gatewayId	Identifies a gateway.
$nodeType	Indicates the device type.
$createTime	Indicates the device creation time.
$lastModifiedTime	Indicates the last modification time.
$deviceInfo	Indicates the device information. For details, see **DeviceInfo structure**.
$services	Indicates the service capabilities of the device. For details, see **DeviceServiceB structure**.

DeviceInfo structure

Parameter	Description
$nodeId	Identifies a device.
$name	Indicates the device name.
$description	Indicates the device description.

Parameter	Description
$manufacturerId	Uniquely identifies a manufacturer. Set this parameter to the value defined in the profile file of the device.
$manufacturerName	Indicates the manufacturer name. Set this parameter to the value defined in the profile file of the device.
$mac	Indicates the MAC address of the device.
$location	Indicates the device location.
$deviceType	Indicates the device type. The upper camel case is used. Set this parameter to the value defined in the profile file of the device, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
$model	Indicates the device model. Set this parameter to the value defined in the profile file of the device. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
$swVersion	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Indicates the firmware version of the device.
$hwVersion	Indicates the hardware version of the device.
$protocolType	Indicates the protocol type used by the device. Set this parameter to the value defined in the profile file of the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
$bridgeId	Identifies the bridge through which the device accesses the IoT platform.
$status	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Indicates the device status. The specific value is determined by the value of **status**. For details, see **status and statusDetail**.
$mute	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Description
$supportedSecurity	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
$isSecurity	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Indicates the signal strength of the device.
$sigVersion	Indicates the SIG version of the device.
$serialNumber	Indicates the serial number of the device.
$batteryLevel	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceServiceB structure

Parameter	Description
$serviceId	Identifies a service.
$reportedProps	Indicates the information reported by the device.
$desiredProps	Indicates the information delivered to the device.
$eventTime	Indicates the time when an event occurs.
$serviceType	Indicates the service type.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.2.7 Modifying Device Shadow Information

Typical Scenario

The IoT platform supports the creation of device shadows. Device shadows store the latest service attribute data reported by devices and service attribute configurations delivered by NAs. (Service attributes are defined in the device profile file.) If the device is offline or abnormal, the NA cannot deliver configuration to the device by issuing commands. In this case, the NA can set the configuration to be delivered to the device shadow. When the device goes online again, the device shadow delivers the configuration to the device. The NA can call this API to modify the configuration information to be delivered to the device on the device shadow.

Each device has only one device shadow, which contains desired and report sections.

- The desired section stores the configurations of device service attributes. If a device is online, the configurations in the desired section are delivered to the device immediately. Otherwise, the configurations in the desired section are delivered to the device when the device goes online.
- The report section stores the latest service attribute data reported by devices. When a device reports data, the IoT platform synchronizes the data to the report section of the device shadow.

API Function

This API is used to modify the configuration information in the desired section of the device shadow. When the device goes online, the configuration information will be delivered to the device.

API Description

```
public function modifyDeviceShadow($mdsInDTO, $deviceId, $appId, $accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$mdsInDTO	Mandatory	body	For details, see **ModifyDeviceShadowInDTO structure**.
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceShadowInDTO

Parameter	Mandatory or Optional	Location	Description
$serviceDesireds	Mandatory	body	Indicates the device configuration or status information to be modified. For details, see **ServiceDesiredDTO**.

ServiceDesiredDTO structure

Parameter	Mandatory or Optional	Location	Description
$serviceId	Optional	body	Identifies a service.
$desired	Optional	body	Indicates the device status.

Return Value

```
response:
Status Code: 200 OK
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100425	The special deviceCapability is not exist.	The device template does not exist. Recommended handling: Check whether the device template has been uploaded to the IoT platform.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100443	The property is forbidden to write.	The device attributes cannot be written.

HTTP Status Code	Error Code	Error Description	Remarks
403	10100 09	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	10100 05	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	10040 3	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	10041 8	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	10002 3	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	10020 3	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.3 Batch Processing

NAs can perform batch operations on devices connected to the IoT platform through the batch processing API.

3.3.3.1 Creating a Batch Task

Typical Scenario

When an NA needs to perform an operation on a batch of devices, the NA can call this API to create a batch task. Currently, the supported batch operations include delivering pending commands to devices in batches.

API Function

This API is used by an NA to create a batch task for devices on the IoT platform.

API Description

```
public function createBatchTask($btcInDTO, $accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$btcInDTO	Mandatory	body	For details, see **BatchTaskCreateInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

BatchTaskCreateInDTO structure

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	body	If the task belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Location	Description
$timeout	Mandatory	body	Indicates the timeout duration of a task, in seconds. Value range: 10-2880.
$taskName	Mandatory	body	Indicates the task name. The value is a string of a maximum of 256 characters.
$taskType	Mandatory	body	Indicates the task type. The values include **DeviceReg**, **DeviceCmd**, **DeviceLocation**, **DeviceMod**, **DeviceDel**, and **DeviceLicense**.
$param	Mandatory	body	Indicates the task parameters. The parameters vary according to **taskType**. For details, see **DeviceCmd**.
$tags	Optional	body	Indicates the tag list. For details, see **TagDTO2**.

DeviceCmd structure

Parameter	Mandatory or Optional	Type	Location	Description
type	mandatory	String	body	Indicates the batch command type. This parameter is mandatory when **taskType** is set to **DeviceCmd**. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	Conditionally mandatory	List<String>	body	Indicates the device ID list. This parameter is mandatory when **type** is set to **DeviceList**.
deviceType	Conditionally mandatory	String	body	Indicates the device type. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
manufacturerId	Conditionally optional	String	body	Identifies the manufacturer. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
model	Conditionally optional	String	body	Indicates the device model. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.

Parameter	Mandatory or Optional	Type	Location	Description
deviceLocation	Conditionally mandatory	String	body	Indicates the location of the device. This parameter is mandatory when **type** is set to **DeviceArea**.
groupList	Conditionally mandatory	List<String>	body	Indicates the group ID list or device group name list. When **type** is set to **GroupIdList**, set this parameter to the group ID. When **type** is set to **GroupList**, set this parameter to the device group name.
command	mandatory	**CommandDTO**	body	Indicates the command information.
callbackUrl	Conditionally optional	String	body	Indicates the push address of the command execution result.
maxRetransmit	Conditionally optional	Integer(0~3)	body	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	Conditionally optional	String	body	Indicates the group tag.

CommandDTO:

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	Mandatory	String(1-128)	body	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.

Parameter	Mandatory or Optional	Type	Location	Description
paras	Optional	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Mandatory or Optional	Location	Description
$tagName	Mandatory	body	Indicates the tag name.
$tagValue	Mandatory	body	Indicates the tag value.

Return Value

BatchTaskCreateOutDTO structure

Parameter	Description
$taskID	Identifies a batch task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	101001	Resource doesn't exist.	The resource does not exist.

HTTP Status Code	Error Code	Error Description	Remarks
200	105001	The batchTask count has reached the limit.	If the number of unfinished tasks is greater than or equal to 10, a message is returned, indicating that the number of tasks reaches the limit.
200	105002	The batchTask name has exist.	The task name exists. Recommended handling: Change the task name.
400	105201	The tagName and tagValue has been used on the platform.	The tagName and tagValue have been used on the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100028	The user has no right.	The user has no operation permission.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105202	The tag is not existed.	The tag does not exist.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.3.2 Querying Information About a Specified Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about the batch task, including the task status and the subtask completion status.

API Function

This API is used by an NA to query the information about a single batch task by task ID.

API Description

```
public function queryOneTask($taskId, $select, $appId, $accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$taskId	Mandatory	path	Identifies a batch task. The value of this parameter is obtained after the batch task is created.
$select	Mandatory	query	Indicates an optional return value. The value can be **tag**. If this parameter is not specified, the value can be **null**.
$appId	Mandatory	query	If the task belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

QueryOneTaskOutDTO

Parameter	Description
$taskId	Identifies a batch task.
$taskName	Indicates the name of the batch task.
$appId	Indicates the **appId** to which the batch task belongs.
$operator	Indicates the operator who delivers the batch task.
$taskFrom	Indicates the source of the batch task. ● **Portal**: The task is created on the SP portal. ● **Northbound**: The task is created by calling a northbound API.
$taskType	Indicates the type of the batch task. The value options are **DeviceReg** and **DeviceCmd**.
$status	Indicates the status of the batch task. The value options are **Pending**, **Running**, **Complete**, and **Timeout**.
$startTime	Indicates the time when the batch task is created.
$timeout	Indicates the time when the batch task times out, in seconds.
$progress	Indicates the progress of the batch task. The value is a permillage ranging from 0 to 1000 and is rounded down.

Parameter	Description
$totalCnt	Indicates the total number of tasks.
$successCnt	Indicates the number of tasks that are successfully executed.
$failCnt	Indicates the number of tasks that fail to be executed.
$timeoutCnt	Indicates the number of tasks with execution timeout.
$expiredCnt	Indicates the number of expired tasks that are not executed.
$completeCnt	Indicates the number of completed tasks, including successful, failed, and timed out tasks.
$successRate	Indicates the task success rate. The value is a permillage ranging from 0 to 1000 and is rounded down.
$param	Indicates the parameters of different types of tasks. For details, see **DeviceCmd structure**.
$tags	Indicates the tag list in the batch task. For details, see **TagDTO2 structure**.

DeviceCmd structure

Parameter	Description
type	Indicates the batch command type. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	Indicates the device ID list. The value is that returned when **type** is set to **DeviceList**.
deviceType	Indicates the type of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
manufacturerId	Identifies a manufacturer. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
model	Indicates the model of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
deviceLocation	Indicates the location of the device. The value is that returned when **type** is set to **DeviceArea**.
groupList	Indicates the group name list. The value is that returned when **type** is set to **GroupList**.
command	Indicates the command information. . For details, see **CommandDTO**.
callbackUrl	Indicates the push address of the command execution result..

Parameter	Description
maxRetransmit	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	Indicates the group tag.

CommandDTO:

Parameter	Description
serviceId	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Description
$tagName	Indicates the tag name.
$tagValue	Indicates the tag value.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100019	Illegal request.	Invalid request. Recommended handling: Check whether the mandatory parameters in the request are set.
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

3.3.3.3 Querying Information About a Subtask of a Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about a subtask of the batch task, including the subtask execution status and subtask content.

API Function

This API is used by an NA to query information about a subtask of a batch task based on specified conditions. This API applies to the current application.

API Description

```
public function queryTaskDetails($qtdInDTO, $accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qtdInDTO	Mandatory	query	For details, see **QueryTaskDetailsInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryTaskDetailsInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Optional	query	If the task belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$taskId	Mandatory	query	Identifies a batch task.
$status	Optional	query	Indicates the task status. The value options are **Pending**, **Success**, **Fail**, and **Timeout**.
$index	Optional	query	Indicates the index in a file. This parameter is used for querying details about a batch device registration task.
$nodeId	Optional	query	Uniquely identifies the device. This parameter is used for querying details about a batch device registration task.

Parameter	Mandatory or Optional	Location	Description
$deviceId	Optional	query	Identifies a device. This parameter is used for querying details about a batch command delivery task.
$commandId	Optional	query	Identifies a command. This parameter is used for querying details about a batch command delivery task.
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.

Return Value

QueryTaskDetailsOutDTO structure

Parameter	Description
$pageNo	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.
$totalCount	Indicates the total number of records.
$taskDetails	Indicates the task details. For details, see **QueryTaskDetailDTO-Cloud2NA structure**.

QueryTaskDetailDTOCloud2NA structure

Parameter	Description
status	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
output	Indicates the output of a batch command delivery task.
error	Indicates the cause of error, in the format of {\"error_code\": \"****\", \"error_desc\":\"*****\"}.
param	Indicates the parameters of different types of tasks. For details, see **ObjectNode**.

ObjectNode:

Parameter	Description
deviceId	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration..
commandId	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.4 Subscription Management

The IoT platform allows NAs to subscribe to device data. If the subscribed device data changes, the IoT platform pushes change notifications to NAs. The subscription management API must be used together with the **Message Push** API.

3.3.4.1 Subscribing to Service Data of the IoT Platform

Typical Scenario

An NA can subscribe to service data of a device on the IoT platform. When the service data changes (for example, the device is registered, the device reports data and the device status changes), the IoT platform can push change notifications to the NA. The NA can call this API to subscribe to different types of service change notifications.

API Function

This API is used by an NA to subscribe to service change notifications on the IoT platform. When the device status or data changes, the IoT platform pushes notifications to the NA.

API Description

```
public function subDeviceData($sddInDTO, $ownerFlag, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$sddInDTO	Mandatory	body	For details, see **SubDeviceDataInDTO structure**.
$ownerFlag	Mandatory	query	Identifies the owner of the callbackUrl. If this parameter is not specified, this parameter can be set to **null**. ● **false**: The callback URL owner is an authorizing application. ● **true**: The callback URL owner is an authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

SubDeviceDataInDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type based on which an NA can receive notification messages pushed by the IoT platform. The value options are as follows: ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceAdded**: device addition; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription ● **commandRsp**: command response; sending such a notification after subscription ● **deviceEvent**: device event; sending such a notification after subscription ● **serviceInfoChanged**: service information change; sending such a notification after subscription ● **deviceModelAdded**: device model addition; sending such a notification after subscription ● **deviceModelDeleted**: device model deletion; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change; sending such a notification after subscription

Parameter	Mandatory or Optional	Location	Description
$callbackUrl	Mandatory	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.
$appId	Optional	body	Identifies the application of the entity that subscribes to a device or rule.
$channel	Optional	body	Indicates the transmission channel. For the MQTT client, the value is **MQTT**. In other cases, the value is **HTTP**.

Response Parameters

Status Code: 201 Created

SubscriptionDTO structure

Parameter	Description
subscriptionId	Identifies a subscription.
notifyType	Indicates the notification type.
callbackUrl	Indicates the callback URL of the subscription.
clientIds	Identifies an MQTT client. The value is that returned when MQTT is subscribed to.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	The request callbackurl is illegal.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
409	100227	The resource is conflicted.	A resource conflict occurs. The notification type has been subscribed to. Recommended handling: Check whether the notification type has been subscribed.

3.3.4.2 Subscribing to Management Data of the IoT Platform

Typical Scenario

An NA can subscribe to management data of a device on the IoT platform. When operations are performed on the device (for example, device upgrade), the IoT platform notifies the NA of the operation status or results. The NA can call this API to subscribe to different types of device upgrade notifications on the IoT platform.

API Function

This API is used by an NA to subscribe to device upgrade notifications on the IoT platform. When the device is upgraded, the IoT platform sends a notification to the NA.

API Description

```
public function subDeviceData2($smdInDTO, $accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$smdInDTO	Mandatory	body	For details, see **SubDeviceManagementDataInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

SubDeviceManagementDataInDTO

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. ● **swUpgradeStateChangeNotify**: software upgrade status change notification; sending such a notification after subscription ● **swUpgradeResultNotify**: software upgrade result notification; sending such a notification after subscription ● **fwUpgradeStateChangeNotify**: hardware upgrade status change notification; sending such a notification after subscription ● **fwUpgradeResultNotify**: hardware upgrade result notification; sending such a notification after subscription
$callbackurl	Mandatory	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.

Response Parameters

```
response:
Status Code: 200 OK
```


Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	Internal server error.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
400	100228	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100229	Get AppKey from header failed.	Failed to obtain the AppKey from the message header.
500	100244	register out route fail.	Failed to register the route. Recommendation: Contact the IoT platform maintenance personnel.

3.3.4.3 Querying a Subscription

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query configuration information about a subscription.

API Function

This API is used by an NA to query the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
public function querySingleSubscription($subscriptionId, $appId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$subscriptionId	Mandatory	path	Identifies a subscription, which is obtained by calling or querying the subscription API.
$appId	Mandatory	query	Identifies the application of the entity that subscribes to a device or rule.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

SubscriptionDTO

Parameter	Description
$subscriptionId	Identifies a subscription.
$notifyType	Indicates the notification type.
$callbackUrl	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

3.3.4.4 Querying Subscription in Batches

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query all subscription configurations of the current application or of a specified subscription type.

API Function

This API is used to query all subscription information of the current application or of a specified subscription type.

API Description

```
public function queryBatchSubscriptions($qbsInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qbsInDTO	Mandatory	query	For details, see **QueryBatchSubInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryBatchSubInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Optional	query	Identifies the application of the entity that subscribes to a device or rule.

Parameter	Mandatory or Optional	Location	Description
$notifyType	Optional	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted:** device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **10**.

Response Parameters

QueryBatchSubOutDTO

Parameter	Description
$totalCount	Indicates the total number of records.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records on each page.
$subscriptions	Indicates the subscription information list. For details, see **SubscriptionDTO structure**.

SubscriptionDTO structure

Parameter	Description
$subscriptionId	Identifies a subscription.
$notifyType	Indicates the notification type.
$callbackUrl	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100224	The resource exceeds 1000, please refinement query conditions.	The number of resources exceeds 1000. Recommended handling: Narrow down the filter criteria.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

3.3.4.5 Deleting a Subscription

Typical Scenario

If an NA does not need to receive a subscription notification message pushed by the IoT platform, the NA can call this API to delete the specified subscription configuration and cancel the subscription.

API Function

This API is used by an NA to delete the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
public function deleteSingleSubscription($subscriptionId, $appId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$subscriptionId	Mandatory	path	Identifies a subscription.
$appId	Optional	query	Identifies the application of the entity that subscribes to a device or rule.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

```
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **subscriptionId** is correct.

3.3.4.6 Deleting Subscriptions in Batches

Typical Scenario

If an NA does not need to receive subscription notification messages pushed by the IoT platform or a specified type of subscription notification messages, the NA can call this API to delete subscription configurations in batches and cancel the subscriptions.

API Function

This API is used to delete all subscriptions, subscriptions of a specified subscription type, or subscriptions of a specified callback URL in batches.

API Description

```
public function deleteBatchSubscriptions($dbsInDTO, $accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$dbsInDTO	Mandatory	body	For details, see **DeleteBatchSubInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeleteBatchSubInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Optional	query	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform. Set this parameter to the value of **appId** of the authorized application.

Parameter	Mandatory or Optional	Location	Description
$notifyType	Optional	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted:** device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification
$callbackUrl	Optional	query	Indicates the callback URL of the subscription.

Parameter	Mandatory or Optional	Location	Description
$channel	Optional	query	Indicates the transmission channel. ● If the value of this parameter is empty, the IoT platform deletes subscriptions with **channel** set to **MQTTS** or **HTTP**. ● If this parameter is set to **MQTTS**, the IoT platform deletes only subscriptions with **channel** set to **MQTTS**. ● If this parameter is set to **HTTP**, the IoT platform deletes only subscriptions with **channel** set to **HTTP**. ● If this parameter is set to other values, subscriptions are not deleted.

Response Parameters

```
Status Code: 204 No Content
```

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **notifyType** is correct.

3.3.5 Message Push

NAs can subscribe to device information from the IoT platform. When the device information changes, the IoT platform pushes change notifications to the NAs. Then, the NAs distribute messages based on the notification type. This API must be used together with the **Subscription Management** API.

3.3.5.1 Pushing Device Registration Notifications

Typical Scenario

After an NA subscribes to device registration notifications (the notification type is **deviceAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when the NA registers a device on the IoT platform by calling the API for registering a directly connected device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device registration notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceAdded(NotifyDeviceAddedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceAddedDTO

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceAdded**.
$deviceId	Mandatory	body	Identifies a device.
$gatewayId	Optional	body	Uniquely identifies a gateway.
$nodeType	Mandatory	body	Indicates the device type. ● **ENDPOINT** ● **GATEWAY** ● **UNKNOWN**
$deviceInfo	Mandatory	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Location	Description
$nodeId	Mandatory	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
$name	Optional	body	Indicates the device name.
$description	Optional	body	Indicates the device description.
$manufacturerId	Optional	body	Uniquely identifies a manufacturer.
$manufacturerName	Optional	body	Indicates the manufacturer name.

Parameter	Mandatory or Optional	Location	Description
$mac	Optional	body	Indicates the MAC address of the device.
$location	Optional	body	Indicates the device location.
$deviceType	Optional	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
$model	Optional	body	Indicates the device model.
$swVersion	Optional	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Optional	body	Indicates the firmware version of the device.
$hwVersion	Optional	body	Indicates the hardware version of the device.
$protocolType	Optional	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
$bridgeId	Optional	body	Identifies the bridge through which the device accesses the IoT platform.
$status	Optional	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Optional	body	Indicates the device status. The specific value is determined by the value of **status**. For details, see **$status and $statusDetail**.
$mute	Optional	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Mandatory or Optional	Location	Description
$supportedSecurity	Optional	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
$isSecurity	Optional	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Optional	body	Indicates the signal strength of the device.
$sigVersion	Optional	body	Indicates the SIG version of the device.
$serialNumber	Optional	body	Indicates the serial number of the device.
$batteryLevel	Optional	body	Indicates the battery level of the device.

$status and $statusDetail

$status	$statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
```



```
    "notifyType":"deviceAdded",
    "deviceId":"*****",
    "gatewayId":"*****",
    "nodeType":"GATEWAY",
    "deviceInfo":{
        "nodeId":"******",
        "name":null,
        "description":null,
        "manufacturerId":null,
        "manufacturerName":null,
        "mac":null,
        "location":null,
        "deviceType":null,
        "model":null,
        "swVersion":null,
        "fwVersion":null,
        "hwVersion":null,
        "protocolType":null,
        "bridgeId":null,
        "status":"OFFLINE",
        "statusDetail":"NOT_ACTIVE",
        "mute":null,
        "supportedSecurity":null,
        "isSecurity":null,
        "signalStrength":null,
        "sigVersion":null,
        "serialNumber":null,
        "batteryLevel":null
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.2 Pushing Device Binding Notifications

Typical Scenario

After an NA subscribes to device binding notifications (the notification type is **bindDevice**) on the IoT platform, the IoT platform sends a notification message to the NA when a directly connected device is connected to the IoT platform and bound to the NA.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device binding notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleBindDevice(NotifyBindDeviceDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyBindDeviceDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **bindDevice**.
$deviceId	Mandatory	body	Identifies a device.
$resultCode	Mandatory	body	Indicates the binding result. The value options are **expired** and **succeeded**.
$deviceInfo	Optional	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Location	Description
$nodeId	Mandatory	body	Identifies a device.
$name	Optional	body	Indicates the device name.
$description	Optional	body	Indicates the device description.
$manufacturerId	Optional	body	Uniquely identifies a manufacturer.
$manufacturerName	Optional	body	Indicates the manufacturer name.
$mac	Optional	body	Indicates the MAC address of the device.
$location	Optional	body	Indicates the device location.
$deviceType	Optional	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Mandatory or Optional	Location	Description
$model	Optional	body	Indicates the device model.
$swVersion	Optional	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Optional	body	Indicates the firmware version of the device.
$hwVersion	Optional	body	Indicates the hardware version of the device.
$protocolType	Optional	body	Indicates the protocol type used by the device. The value options are **Z-Wave**, **ZigBee**, and **WPS**.
$bridgeId	Optional	body	Identifies the bridge through which the device accesses the IoT platform.
$status	Optional	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Optional	body	Indicates the device status. The specific value is determined by the value of **status**. For details, see **$status and $statusDetail**.
$mute	Optional	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
$supportedSecurity	Optional	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.

Parameter	Mandatory or Optional	Location	Description
$isSecurity	Optional	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Optional	body	Indicates the signal strength of the device.
$sigVersion	Optional	body	Indicates the SIG version of the device.
$serialNumber	Optional	body	Indicates the serial number of the device.
$batteryLevel	Optional	body	Indicates the battery level of the device.

$status and $statusDetail

$status	$statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"bindDevice",
    "deviceId":"*****",
    "resultCode":"succeeded",
    "deviceInfo":{
        "name":"Sensor_12",
```



```
        "manufacturer":"wulian",
        "deviceType":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
        "fwVersion": "..."
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smockdetector",
        "nodeType":"GATEWAY"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.3 Pushing Device Information Change Notifications

Typical Scenario

After an NA subscribes to device information change notifications (the notification type is **deviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device configuration or status (such as manufacturer, location, version and online status) changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device information change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceInfoChanged(NotifyDeviceInfoChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceInfoChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceInfoChanged**.
$deviceId	Mandatory	body	Identifies a device.
$gatewayId	Mandatory	body	Uniquely identifies a gateway.
$deviceInfo	Mandatory	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Location	Description
$nodeId	Mandatory	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
$name	Optional	body	Indicates the device name.
$description	Optional	body	Indicates the device description.
$manufacturerId	Optional	body	Uniquely identifies a manufacturer.
$manufacturerName	Optional	body	Indicates the manufacturer name.
$mac	Optional	body	Indicates the MAC address of the device.
$location	Optional	body	Indicates the device location.

Parameter	Mandatory or Optional	Location	Description
$deviceType	Optional	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
$model	Optional	body	Indicates the device model.
$swVersion	Optional	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Optional	body	Indicates the firmware version of the device.
$hwVersion	Optional	body	Indicates the hardware version of the device.
$protocolType	Optional	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
$bridgeId	Optional	body	Identifies the bridge through which the device accesses the IoT platform.
$status	Optional	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Optional	body	Indicates the device status. The specific value is determined by the value of **status**. For details, see **$status and $statusDetail**.
$mute	Optional	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.

Parameter	Mandatory or Optional	Location	Description
$supportedSecurity	Optional	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
$isSecurity	Optional	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Optional	body	Indicates the signal strength of the device.
$sigVersion	Optional	body	Indicates the SIG version of the device.
$serialNumber	Optional	body	Indicates the serial number of the device.
$batteryLevel	Optional	body	Indicates the battery level of the device.

$status and $statusDetail

$status	$statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
```



```
    "notifyType ":"deviceInfoChanged",
    "deviceId":"******************************",
    "gatewayId":"*************",
    "deviceInfo":{
        "name":"Sensor_12",
        "manufacturer":"wulian",
        "type":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
        "fwVersion": "..."
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smock detector"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.4 Pushing Device Data Change Notifications

Typical Scenario

After an NA subscribes to device data change notifications (the notification type is **deviceDataChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of a single service attribute.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device data change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceDataChanged(NotifyDeviceDataChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDataChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceDataChanged**.
$requestId	Optional	body	Indicates the sequence number of the message, which uniquely identifies the message.
$deviceId	Mandatory	body	Identifies a device.
$gatewayId	Mandatory	body	Uniquely identifies a gateway.
$service	Mandatory	body	Indicates service data of the device. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Location	Description
$serviceId	Mandatory	body	Identifies a service.
$serviceType	Mandatory	body	Indicates the service type.
$data	Mandatory	body	Indicates service data information.
$eventTime	Mandatory	body	Indicates the event occurrence time in the format of yyyymmddThhmissZ, for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
```



```
Body:
{
    "notifyType":"deviceDataChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":{
        "serviceId":"Brightness",
        "serviceType":"Brightness",
        "data":{
            "brightness":80
        },
        "eventTime":"20170311T163657Z"
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.5 Pushing Batch Device Data Change Notifications

Typical Scenario

After an NA subscribes to batch device data change notifications (the notification type is **deviceDatasChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of multiple service attributes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to batch device data change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceDatasChanged(NotifyDeviceDatasChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDatasChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceDatasChanged**.
$requestId	Mandatory	body	Indicates the sequence number of the message, which uniquely identifies the message.
$deviceId	Mandatory	body	Identifies a device.
$gatewayId	Mandatory	body	Uniquely identifies a gateway.
$services	Mandatory	body	Indicates a service list. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Location	Description
$serviceId	Mandatory	body	Identifies a service.
$serviceType	Mandatory	body	Indicates the service type.
$data	Mandatory	body	Indicates service data information.
$eventTime	Mandatory	body	Indicates the time when an event is reported in the format of yyyyMMddTHHmmssZ, for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDatasChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":[
        {
            "serviceId":"Brightness",
```



```
            "serviceType":"Brightness",
            "data":{
                "brightness":80
            },
            "eventTime":"20170311T163657Z"
        },
        {
            "serviceId":"Color",
            "serviceType":"Color",
            "data":{
                "value":"red"
            },
            "eventTime":"20170311T163657Z"
        }
    ]
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.6 Pushing Device Service Information Change Notifications

Typical Scenario

After an NA subscribes to device service information change notifications (the notification type is **serviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device to modify the device service information.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service information change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleServiceInfoChanged(NotifyServiceInfoChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyServiceInfoChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **serviceInfoChanged**.
$deviceId	Mandatory	body	Identifies a device.
$gatewayId	Mandatory	body	Uniquely identifies a gateway.
$serviceId	Mandatory	body	Identifies a service.
$serviceType	Mandatory	body	Indicates the service type.
$serviceInfo	Mandatory	body	Indicates the device service information, which is incrementally reported. For details, see **ServiceInfo structure**.

ServiceInfo structure

Parameter	Mandatory or Optional	Location	Description
$muteCmds	Optional	body	Indicates the device command list.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"serviceInfoChanged",
    "deviceId":"*******",
    "serviceId":"*******",
    "serviceType":"*******",
    "gatewayId":"*******",
    "serviceInfo":{
        "muteCmds":"VIDEO_RECORD"
```

```
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.7 Pushing Device Deletion Notifications

Typical Scenario

After an NA subscribes to device deletion notifications (the notification type is **deviceDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when the device is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device deletion notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceDeleted(NotifyDeviceDeletedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDeletedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceDeleted**.
$deviceId	Mandatory	body	Identifies a device.

Parameter	Mandatory or Optional	Location	Description
$gatewayId	Mandatory	body	Uniquely identifies a gateway.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDeleted",
    "deviceId":"******************************",
    "gatewayId":"*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.8 Pushing Device Acknowledgment Notifications

Typical Scenario

After an NA subscribes to device acknowledgment notifications (the notification type is **messageConfirm**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command acknowledgment message (for example, the command is delivered or executed).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device acknowledgment notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleMessageConfirm(NotifyMessageConfirmDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyMessageConfirmDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **messageConfirm**.
$header	Mandatory	body	For details, see **MessageConfirmHeader structure**.
$body	Mandatory	body	Based on the service definition, an acknowledgment message can carry information such as status change.

MessageConfirmHeader structure

Parameter	Mandatory or Optional	Location	Description
$requestId	Mandatory	body	Indicates the sequence number of the message, which uniquely identifies the message.
$from	Mandatory	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**
$to	Mandatory	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.

Parameter	Mandatory or Optional	Location	Description
$status	Mandatory	body	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been received. ● **executed**: The command has been executed.
$timestamp	Mandatory	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"messageConfirm",
    "header":{
        "requestId":"************",
        "from":"************",
        "to":"************",
        "status":"delivered",
        "timestamp":"20151212T121212Z"
    },
    "body":{

    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.9 Pushing Device Command Response Notifications

Typical Scenario

After an NA subscribes to device command response notifications (the notification type is **commandRsp**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command response message (for example, the command execution succeeds or fails).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device command response notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleCommandRsp(NotifyCommandRspDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyCommandRspDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **commandRsp**.
$header	Mandatory	body	For details, see **CommandRspHeader structure**.
$body	Mandatory	body	Indicates the content of the message body of the response command.

CommandRspHeader structure

Parameter	Mandatory or Optional	Location	Description
$requestId	Mandatory	body	Indicates the sequence number of the message, which uniquely identifies the message.
$from	Mandatory	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**
$to	Mandatory	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
$deviceId	Mandatory	body	Identifies a device.
$serviceType	Mandatory	body	Indicates the type of the service to which a command belongs.
$method	Mandatory	body	Indicates a stored response command, for example, **INVITE-RSP**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"commandRsp",
    "header":{
        "requestId":"***********",
        "from":"***********",
        "to":"***********",
        "deviceId":"***********",
        "serviceType":"Camera",
        "method":"MUTE_COMMANDS"
    },
    "body":{

    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```


3.3.5.10 Pushing Device Event Notifications

Typical Scenario

After an NA subscribes to device event notifications (the notification type is **deviceEvent**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform receives the event message reported by the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device event notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceEvent(NotifyDeviceEventDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceEventDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceEvent**.
$header	Mandatory	body	For details, see **DeviceEventHeader structure**.
$body	Mandatory	body	Indicates the content of the event message.

DeviceEventHeader structure

Parameter	Mandatory or Optional	Location	Description
$eventType	Mandatory	body	Indicates the event type.
$from	Mandatory	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**
$timestamp	Mandatory	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
$eventTime	Mandatory	body	Indicates the time when an event is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
$deviceId	Mandatory	body	Identifies a device.
$serviceType	Mandatory	body	Indicates the type of the service to which a command belongs.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceEvent",
    "header":{
        "eventType":"*******",
        "from":"/devices/{deviceId}/services/{serviceId}",
        "deviceId":"*******",
        "serviceType":"*******",
        "timestamp":"20151212T121212Z",
        "eventTime":"20151212T121212Z"
    },
    "body":{
        "usedPercent":80
    }
}
```


Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.11 Pushing Device Model Addition Notifications

Typical Scenario

After an NA subscribes to device model addition notifications (the notification type is **deviceModelAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is added on the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model addition notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceModelAdded(NotifyDeviceModelAddedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceModelAddedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceModelAdded**.

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceType	Mandatory	body	Indicates the device type.
$manufacturerName	Mandatory	body	Indicates the operator name.
$manufacturerId	Mandatory	body	Identifies the operator.
$model	Mandatory	body	Indicates the device model.
$protocolType	Mandatory	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceType":"*************",
    "manufacturerName":"wulian",
    " manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"zigbee"
}
```

Response Example

```
Response:
Status Code: 200 OK
```


3.3.5.12 Pushing Device Model Deletion Notifications

Typical Scenario

After an NA subscribes to device model deletion notifications (the notification type is **deviceModelDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model deletion notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleDeviceModelDeleted(NotifyDeviceModelDeletedDTO body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceModelDeletedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceModelDeleted**.
$appId	Mandatory	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.

Parameter	Mandatory or Optional	Location	Description
$deviceType	Mandatory	body	Indicates the device type.
$manufacturerName	Mandatory	body	Indicates the operator name.
$manufacturerId	Mandatory	body	Identifies the operator.
$model	Mandatory	body	Indicates the device model.
$protocolType	Mandatory	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceTyp ":"*************",
    " manufacturerName":"*************",
    "manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"*************"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.13 Pushing Device Shadow Status Change Notifications

Typical Scenario

After an NA subscribes to device shadow status change notifications (the notification type is **deviceDesiredPropertiesModifyStatusChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device shadow on the IoT platform succeeds or fails to synchronize data to the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device shadow status change notifications.

Note

1. When **subscribing to platform service data**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	funciton handleDeviceDesiredStatusChanged(NotifyDeviceDesiredStatusChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyDeviceDesiredStatusChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **deviceDesiredPropertiesModifyStatusChanged**.
$deviceId	Mandatory	body	Identifies a device.
$serviceId	Mandatory	body	Identifies a service.
$properties	Mandatory	body	Indicates data attributes of a device shadow.
$status	Mandatory	body	Indicates the status. The value options are **DELIVERED** and **FAILED**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDesiredPropertiesModifyStatusChanged",
    "deviceId":"******",
    "serviceId":"Device",
    "properties":{
        "Model Number":1,
        "Serial Number":2,
        "Firmware Version":"v1.1.0"
    },
    "status":"DELIVERED"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.14 Pushing Software Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to software upgrade status change notifications (the notification type is **swUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the software upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade status change notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver

Callback API	function handleSwUpgradeStateChanged(NotifySwUpgradeStateChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifySwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **swUpgradeStateChangeNotify**.
$deviceId	Mandatory	body	Identifies a device.
$appId	Mandatory	body	Identifies the application to which the device belongs.
$operationId	Mandatory	body	Identifies a software upgrade task.
$subOperationId	Mandatory	body	Identifies a software upgrade sub-task.
$swUpgradeState	Mandatory	body	Indicates the software upgrade status. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
```



```
    "notifyType":"swUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"**********",
    "subOperationId":"**********",
    "swUpgradeState":"downloading"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.15 Pushing Software Upgrade Result Notifications

Typical Scenario

After an NA subscribes to software upgrade result change notifications (the notification type is **swUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a software upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade result notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleSwUpgradeResult(NotifySwUpgradeResultDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifySwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **swUpgradeResultNotify**.
$deviceId	Mandatory	body	Identifies a device.
$appId	Mandatory	body	Identifies the application to which the device belongs.
$operationId	Mandatory	body	Identifies a software upgrade task.
$subOperationId	Mandatory	body	Identifies a software upgrade sub-task.
$curVersion	Mandatory	body	Indicates the current software version of the device.
$targetVersion	Mandatory	body	Indicates the target software version to which the device is to be upgraded.
$sourceVersion	Mandatory	body	Indicates the source software version of the device.
$swUpgradeResult	Mandatory	body	Indicates the software upgrade result. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
$upgradeTime	Mandatory	body	Indicates the upgrade time.
$resultDesc	Mandatory	body	Indicates the upgrade result description.
$errorCode	Mandatory	body	Indicates a status error code reported by the device.
$description	Mandatory	body	Indicates the description of the cause of error.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
```



```
Body:
{
    "notifyType":"swUpgradeResultNotify",
    "deviceId":"******",
    "appId":"**********",
    "operationId":"**********",
    "subOperationId":"**********",
    "curVersion":"1.3",
    "targetVersion":"1.5",
    "sourceVersion":"1.0",
    "swUpgradeResult":"SUCCESS",
    "upgradeTime":"***",
    "resultDesc":"***",
    "errorCode":"***",
    "description":"***"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.16 Pushing Firmware Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to firmware upgrade status change notifications (the notification type is **fwUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the firmware upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade status change notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleFwUpgradeStateChanged(NotifyFwUpgradeStateChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyFwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **fwUpgradeStateChangeNotify**.
$deviceId	Mandatory	body	Identifies a device.
$appId	Mandatory	body	Identifies the application to which the device belongs.
$operationId	Mandatory	body	Identifies a firmware upgrade task.
$subOperationId	Mandatory	body	Identifies a firmware upgrade sub-task.
$step	Mandatory	body	Indicates the firmware upgrade status. The value options are **0**, **1**, **2**, and **3**.
$stepDesc	Mandatory	body	Indicates the upgrade status description. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"**************",
    "subOperationId":"***********",
    "step":"1",
    "stepDesc":"downloading"
}
```


Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.17 Pushing Firmware Upgrade Result Notifications

Typical Scenario

After an NA subscribes to firmware upgrade result change notifications (the notification type is **fwUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a firmware upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade result notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	function handleFwUpgradeResult(NotifyFwUpgradeResultDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyFwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Location	Description
$notifyType	Mandatory	body	Indicates the notification type. The value is **fwUpgradeResultNotify**.

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	body	Identifies a device.
$appId	Mandatory	body	Identifies the application to which the device belongs.
$operationId	Mandatory	body	Identifies a firmware upgrade task.
$subOperationId	Mandatory	body	Identifies a firmware upgrade sub-task.
$curVersion	Mandatory	body	Indicates the current firmware version of the device.
$targetVersion	Mandatory	body	Indicates the target firmware version to which the device is to be upgraded.
$sourceVersion	Mandatory	body	Indicates the source firmware version of the device.
$Status	Mandatory	body	Indicates the upgrade result. ● **SUCCESS** ● **FAIL**
$statusDesc	Mandatory	body	Indicates the upgrade result description. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
$upgradeTime	Mandatory	body	Indicates the firmware upgrade duration.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeResultNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"***********",
    "subOperationId":"***********",
```

```
    "curVersion":"1.6",
    "targetVersion":"1.6",
    "sourceVersion":"1.3",
    "status":"SUCCESS",
    "statusDesc":"****",
    "upgradeTime":"****"
}
```

Response Example

```
Response:
Status Code: 200 OK
```

3.3.5.18 NB-IoT Device Command Status Change Notification

Typical Scenario

When an NA creates a device command with the callback URL specified, the IoT platform pushes a notification message to the NA if the command status changes (failed, successful, timeout, sent, or delivered).

API Function

The IoT platform pushes notification messages to NAs when the command status changes.

Note

1. When **Creating Device Commands**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver/cmd
Callback API	function handleNBCommandStateChanged(NotifyNBCommandStatusChangedDTO $body)
Class	PushMessageReceiver

Parameter Description

NotifyNBCommandStatusChangedDTO structure

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	body	Uniquely identifies a device.

Parameter	Mandatory or Optional	Location	Description
$commandId	Mandatory	body	Identifies a command, which is generated by the IoT platform during device creation.
$result	Mandatory	body	For details, see **NBCommandResult structure**.

NBCommandResult structure

Parameter	Mandatory or Optional	Location	Description
$resultCode	Mandatory	body	Indicates the command status result. ● **SENT**: The IoT platform has delivered a command to the device but has not received a response from the device. ● **DELIVERED**: The IoT platform receives a response from a device. ● **SUCCESS**: The IoT platform receives a command result and the result is success. ● **FAIL**: The IoT platform receives a command result and the result is a failure.
$resultDetail	Mandatory	body	Indicates the user-defined fields carried in the command result.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
Request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "deviceId":"*********",
    "commandId":"***********",
    "result":{
        "resultCode":"DELIVERED",
        "resultDetail":null
    }
}
```

Response Example

```
Response:
Status Code: 200 OK
```


3.3.6 Command Delivery (NB-IoT Commands)

3.3.6.1 Creating Device Commands

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform provides two command delivery modes:

- Immediate delivery: The IoT platform delivers commands to devices immediately after receiving the commands. This ensures real-time performance but does not ensure serialization.
- Pending delivery: After receiving commands, the IoT platform caches the commands. When the devices are reachable, the IoT platform delivers the commands in sequence. Specifically, the IoT platform delivers the latter command only after receiving the response of the previous command (which is the ACK automatically replied by the module) to ensure serialization instead of real-time performance.

API Function

This API is used by NAs to deliver commands to devices. Immediate delivery and pending delivery are supported on the IoT platform.

API Description

```
public function postDeviceCommand($pdcInDTO, $appId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$pdcInDTO	Mandatory	body	For details, see **PostDeviceCommandInDTO structure**.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

PostDeviceCommandInDTO

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	body	Uniquely identifies the device that delivers the command.
$command	Mandatory	body	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Optional	body	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Optional	body	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2). If this parameter is set to **0**, the IoT platform will deliver the command to the specific device immediately regardless of the command mode set on the IoT platform (if the device is sleeping or the link has aged, the device cannot receive the command, the IoT platform cannot receive any response from the device, and the command times out in the end).
$maxRetransmit	Optional	body	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Location	Description
$serviceId	Mandatory	body	Identifies the service corresponding to the command. The value of this parameter must be the same as **serviceId** defined in the profile.

Parameter	Mandatory or Optional	Location	Description
$method	Mandatory	body	Indicates the command name. The value of this parameter must be the same as the command name defined in the profile file.
$paras	Mandatory	ObjectNode	body Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device. If no parameter is defined in the command in the profile file, left it blank, that is, "paras": {}.

Response Parameters

PostDeviceCommandOutDTO

Parameter	Description
$commandId	Identifies a device command.
$appId	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceId	Uniquely identifies the device to which the command is delivered.
$command	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).

Parameter	Description
$status	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
$result	Indicates the detailed command execution result.
$creationTime	Indicates the time when the command is created.
$executeTime	Indicates the time when the command is executed.
$platformIssuedTime	Indicates the time when the IoT platform sends the command.
$deliveredTime	Indicates the time when the command is delivered.
$issuedTimes	Indicates the number of times the IoT platform delivers the command.
$maxRetransmit	Indicates the maximum number of times the command can be retransmitted.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100223	Command counts has reached the upLimit.	The number of cached commands reaches the limit. The number of commands in the **PENDING** state does not exceed the limit. The default value is **20**. Recommended handling: If the commands cached on the IoT platform need to be executed, enable the device to report data to trigger delivery of the cache commands. If a command cached on the IoT platform does not need to be executed, call the API used for modifying device commands V4 to change the state of the command from **PENDING** to **CANCELED**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100612	Device is zombie.	The device is a zombie device. (The interval between the current system time and the time when the device went online exceeds the threshold. The default value is seven days.) Recommended handling: Run the command again after the device goes online.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.6.2 Querying Device Commands

Typical Scenario

After an NA delivers a command to a device, the NA can call this API to query the status and content of the delivered command on the IoT platform to check the command execution status.

API Function

This API is used by an NA to query the status and content of delivered commands on the IoT platform. All the commands delivered by the current application in a specified period or all the commands delivered to a specified device can be queried.

API Description

```
public function queryDeviceCommand($qdcInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qdcInDTO	Mandatory	query	For details, see **QueryDeviceCommandInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCommandInDTO

Parameter	Mandatory or Optional	Location	Description
$pageNo	Optional	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
$pageSize	Optional	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
$deviceId	Optional	query	Identifies the device whose commands are to be queried.
$startTime	Optional	query	Indicates the start time. Commands delivered later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	query	Indicates the end time. Commands delivered earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$appId	Optional	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCommandOutDTO

Parameter	Description
$pagination	Indicates pagination information. For details, see **Pagination structure**.
$data	Indicates the device command list. For details, see **DeviceCommandRespV4 structure**.

Pagination structure

Parameter	Description
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records to be displayed on each page.

Parameter	Description
$totalSize	Indicates the total number of records.

DeviceCommandRespV4 structure

Parameter	Description
$commandId	Identifies a device command.
$appId	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceId	Uniquely identifies the device to which the command is delivered.
$command	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
$status	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
$result	Indicates the detailed command execution result.
$creationTime	Indicates the time when the command is created.
$executeTime	Indicates the time when the command is executed.
$platformIssuedTime	Indicates the time when the IoT platform sends the command.
$deliveredTime	Indicates the time when the command is delivered.
$issuedTimes	Indicates the number of times the IoT platform delivers the command.

Parameter	Description
$maxRetransmit	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Description
$serviceId	Mandatory	Identifies the service corresponding to the command.
$method	Mandatory	Indicates the command name.
$paras	Optional	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: ● Ensure that neither **startTime** nor **endTime** is null and the value of **endTime** is later than that of **startTime**. ● Ensure that **pageNo** is not null and the value of **pageNo** is greater than **0**. ● Ensure that **pageSize** is not null and the value of **pageSize** is greater than **1**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.6.3 Modifying Device Commands

Typical Scenario

NAs can call this API to modify the status of commands that have not been canceled, expired, or executed. Currently, the status of such commands can only be changed to **Canceled**.

API Function

This API is used by NAs to modify the status of commands. Currently, the status of such commands can only be changed to **Canceled**. That is, the commands are revoked.

API Description

```
public function updateDeviceCommand($udcInDTO, $deviceCommandId, $appId,
$accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$udcInDTO	Mandatory	body	For details, see **UpdateDeviceCommandInDTO structure**.
$deviceCommandId	Mandatory	path	Identifies the command whose status is to be modified. The value of this parameter is obtained after the API used for creating device commands is called.

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

UpdateDeviceCommandInDTO

Parameter	Mandatory or Optional	Description
$status	Mandatory	Indicates the command execution result. The value can be **CANCELED**, which indicates that the command is revoked.

Response Parameters

UpdateDeviceCommandOutDTO

Parameter	Description
$commandId	Identifies a device command.
$appId	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceId	Uniquely identifies the device to which the command is delivered.
$command	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).

Parameter	Description
$status	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
$result	Indicates the detailed command execution result.
$creationTime	Indicates the time when the command is created.
$executeTime	Indicates the time when the command is executed.
$platformIssuedTime	Indicates the time when the IoT platform sends the command.
$deliveredTime	Indicates the time when the command is delivered.
$issuedTimes	Indicates the number of times the IoT platform delivers the command.
$maxRetransmit	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Location	Description
$serviceId	Mandatory	body	Identifies the service corresponding to the command.
$method	Mandatory	body	Indicates the command name.
$paras	Optional	body	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
200	100434	The device command is not existed.	The device command does not exist. Recommended handling: Check whether the device command ID in the request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
200	100435	The device command already canceled, expired or executed, Cannot cancel.	The device command has been canceled, expired, or executed. It cannot be canceled.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.6.4 Creating Device Command Revocation Tasks

Typical Scenario

After an NA delivers commands to a device, the IoT platform does not deliver the commands to the device for execution (the commands are in the **DEFAULT** state) if the commands are in queue or the device is offline. In this case, the NA can call this API to revoke all the undelivered commands of a specified device. Commands that have been delivered cannot be revoked.

API Function

This API is used by an NA to create a command revocation task to revoke all undelivered commands (that is, commands in the **DEFAULT** state) with the specified device ID on the IoT platform.

API Description

```
public function createDeviceCmdCancelTask($cdcctInDTO, $appId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$cdcctInDTO	Mandatory	body	For details, see **CreateDeviceCmdCancelTaskInDTO structure**.
$appId	Mandatory	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Location	Description
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateDeviceCmdCancelTaskInDTO

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	body	Identifies the device whose commands are to be revoked. The revocation task will revoke all commands delivered to this device.

Response Parameters

CreateDeviceCmdCancelTaskOutDTO

Parameter	Description
$taskId	Identifies a command revocation task.
$appId	Identifies the application to which the command revocation task belongs.
$deviceId	Identifies the device whose commands are to be revoked by the revocation task.
$status	Indicates the status of the command revocation task. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
$totalCount	Indicates the total number of revoked commands.
$deviceCommands	Indicates the revoked device command list. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Description
$commandId	Identifies a device command.
$appId	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceId	Uniquely identifies the device to which the command is delivered.
$command	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
$status	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
$result	Indicates the detailed command execution result.
$creationTime	Indicates the time when the command is created.
$executeTime	Indicates the time when the command is executed.
$platformIssuedTime	Indicates the time when the IoT platform sends the command.
$deliveredTime	Indicates the time when the command is delivered.
$issuedTimes	Indicates the number of times the IoT platform delivers the command.
$maxRetransmit	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Description
$serviceId	Mandatory	Identifies the service corresponding to the command.
$method	Mandatory	Indicates the command name.
$paras	Optional	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.6.5 Querying Command Revocation Tasks

Typical Scenario

After delivering a command revocation command to a device, an NA can call this API to query the execution status of the command revocation task.

API Function

This API is used by an NA to query the information and status of one or more command revocation tasks based on specified conditions on the IoT platform.

API Description

```
public function queryDeviceCmdCancelTask($qdcctInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qdcctInDTO	Mandatory	query	For details, see **QueryDeviceCmdCancelTaskInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCmdCancelTaskInDTO

Parameter	Mandatory or Optional	Location	Description
$pageNo	Optional	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
$pageSize	Optional	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
$taskId	Optional	query	Identifies a command revocation task.
$deviceId	Optional	query	Identifies the device whose commands are to be revoked by the revocation task.
$status	Optional	query	Indicates the status of the command revocation task.
$startTime	Optional	query	Indicates the start time. Revocation tasks created later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	query	Indicates the end time. Revocation tasks created earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Location	Description
$appId	Optional	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCmdCancelTaskOutDTO

Parameter	Description
$pagination	Indicates pagination information. For details, see **Pagination structure**.
$data	Indicates the device command list. For details, see **CreateDeviceCmdCancelTaskOutDTO structure**.

Pagination structure

Parameter	Description
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records to be displayed on each page.
$totalSize	Indicates the total number of records, that is, the total number of commands queried in the command revocation task.

CreateDeviceCmdCancelTaskOutDTO structure

Parameter	Description
$taskId	Identifies a command revocation task.
$appId	Identifies the application to which the command revocation task belongs.
$deviceId	Identifies the device whose commands are to be revoked by the revocation task.

Parameter	Description
$status	Indicates the status of the command revocation task. ● **WAITING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
$totalCount	Indicates the total number of revoked commands.
$deviceCommands	Indicates a list of device commands to be revoked by the revocation task. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Description
$commandId	Identifies a device command.
$appId	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
$deviceId	Uniquely identifies the device to which the command is delivered.
$command	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
$callbackUrl	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
$expireTime	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
$status	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.

Parameter	Description
$result	Indicates the detailed command execution result.
$creationTime	Indicates the time when the command is created.
$executeTime	Indicates the time when the command is executed.
$platformIssuedTime	Indicates the time when the IoT platform sends the command.
$deliveredTime	Indicates the time when the command is delivered.
$issuedTimes	Indicates the number of times the IoT platform delivers the command.
$maxRetransmit	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Description
$serviceId	Mandatory	Identifies the service corresponding to the command.
$method	Mandatory	Indicates the command name.
$paras	Optional	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.7 Command Delivery (Non-NB-IoT Commands)

3.3.7.1 Calling Device Services

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform does not cache commands but delivers commands directly. When a device is offline, the commands fail to be delivered. The formats of the delivered command need to be defined by the NAs and devices. The IoT platform encapsulates and transparently transmits the commands over this API.

NOTE

Currently, this API can be used to deliver commands only to gateways equipped with the IoT Agent or AgentLite to control non-directly connected devices under the gateways.

API Function

This API is used to immediately deliver commands to gateways equipped with the IoT Agent or AgentLite to control the gateways. This API applies to devices registered with the current application.

API Description

```
public function invokeDeviceService($deviceId, $serviceId, $commandDTO, $appId,
$accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	path	Uniquely identifies a device.
$serviceId	Mandatory	path	Uniquely identifies a service.
$commandDTO	Mandatory	body	For details, see **CommandDTO structure**.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CommandDTO

Parameter	Mandatory or Optional	Location	Description
$header	Mandatory	body	For details, see **CommandNA2CloudHeader structure**.
$body	Optional	body	Indicates the message body. The content of the JsonObject is a list of key-value pairs. Every key is the **paraName** of a command defined in the profile.

CommandNA2CloudHeader structure

Parameter	Mandatory or Optional	Location	Description
$requestId	Optional	body	Identifies a command. The value of this parameter must be unique.

Parameter	Mandatory or Optional	Location	Description
$mode	Mandatory	body	Indicates whether an ACK message is required. ● **NOACK**: No ACK message is required. ● **ACK**: An ACK message is required. ● Other values: invalid
$from	Optional	body	Indicates the address of the message sender. ● Request initiated by an application: **/users/{userId}** ● Request initiated by an NA: **/{serviceName}** ● Request initiated by the IoT platform: **/cloud/{serviceName}**
$toType	Optional	body	Indicates the type of the message recipient. The value options are **CLOUD** and **GATEWAY**.
$to	Optional	body	Indicates the address of the message recipient.
$method	Mandatory	body	Indicates the command name. For example, a **DISCOVERY** command is used to discover non-directly connected devices, and a **REMOVE** command is used to delete non-directly connected devices.
$callbackURL	Optional	body	Indicates the callback URL.

Response Parameters

InvokeDeviceServiceOutDTO

Parameter	Description
$status	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been delivered. This value is returned when **toType** is set to **CLOUD**. ● **failed**: The command fails to be delivered. This value is returned when **toType** is set to **CLOUD**.
$timestamp	Indicates the timestamp used for sending a command. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Description
$requestId	Identifies a device command. ● When **toType** is set to **GATEWAY**, if **requestId** is carried in a request, the response carries the same **requestId** as the request; if **requestId** is not carried in a request, the IoT platform allocates a sequence number for the response. ● When **toType** is set to **CLOUD**, the value of this parameter is null.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the gateway is normal.

HTTP Status Code	Error Code	Error Description	Remarks
200	100432	The device command is muted.	The device command is muted. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	102203	CommandName is invalid.	The command name is invalid. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
403	100450	The gateway is not online.	The gateway is offline. Recommended handling: Check whether the connection between the gateway and the IoT platform is normal.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100444	The serviceType is not exist.	The service type does not exist. Recommended handling: Check whether the service type carried in the API request parameter **toType** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

3.3.8 Data Collection

The IoT platform allows NAs to query basic information about a device and view historical data reported by the device by hour, day, or month.

3.3.8.1 Querying Information About a Device

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of a device that has been registered with the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information of a specified device based on the device ID on the IoT platform, such as configuration, status and service attributes.

API Description

```
public function querySingleDeviceInfo($deviceId, $select, $appId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$deviceId	Mandatory	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
$select	Optional	query	Indicates the query condition. The value can be **IMSI**.

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QuerySingleDeviceInfoOutDTO

Parameter	Description
$deviceId	Identifies a device.
$gatewayId	Identifies a gateway.
$nodeType	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
$createTime	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$lastModifiedTime	Indicates the last time when the device is modified.
$deviceInfo	Indicates information about the device. For details, see **DeviceInfo structure**.
$services	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Description
$nodeId	Identifies a device.
$name	Indicates the device name.
$description	Indicates the device description.
$manufacturerId	Uniquely identifies a manufacturer.
$manufacturerName	Indicates the manufacturer name.
$mac	Indicates the MAC address of the device.

Parameter	Description
$location	Indicates the device location.
$deviceType	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
$model	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
$swVersion	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Indicates the firmware version of the device.
$hwVersion	Indicates the hardware version of the device.
$protocolType	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
$bridgeId	Identifies the bridge through which the device accesses the IoT platform.
$status	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.
$mute	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
$supportedSecurity	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
$isSecurity	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Indicates the signal strength of the device.
$sigVersion	Indicates the SIG version of the device.
$serialNumber	Indicates the serial number of the device.

Parameter	Description
$batteryLevel	Indicates the battery level of the device.

status and statusDetail

$status	$statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Description
$serviceId	Identifies a service.
$serviceType	Indicates the service type.
$serviceInfo	Indicates service information of the device. For details, see **ServiceInfo structure**.
$data	Indicates an attribute-value pair (AVP).
$eventTime	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Description
$muteCmds	Indicates the device command list.

Error Codes

HTTP Status Code	Error Codes	Error Description	Remarks
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.

HTTP Status Code	Error Codes	Error Description	Remarks
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.8.2 Querying a Device Information List

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of multiple devices that have been registered with the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information (such as configuration, status and service attributes) of multiple devices based on specified conditions on the IoT platform.

API Description

```
public function queryBatchDevicesInfo($qbdiInDTO, $accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qbdiInDTO	Mandatory	query	For details, see **PostDeviceCommandInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

PostDeviceCommandInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$gatewayId	Optional	query	Identifies a gateway.
$nodeType	Optional	query	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
$deviceType	Optional	query	Indicates the device type.
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the number of records on each page. The default value is **1**.
$status	Optional	query	Indicates the device status. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline. ● **ABNORMAL**: The device is abnormal.

Parameter	Mandatory or Optional	Location	Description
$startTime	Optional	query	Indicates the start time. Records with the device registration time later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	query	Indicates the end time. Records with the device registration time earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$sort	Optional	query	Indicates the sorting mode of queried records. ● **ASC**: Records are sorted in ascending order of device registration time. ● **DESC**: Records are sorted in descending order of device registration time. The default value is **DESC**.
$select	Optional	query	Indicates the record to be returned. The value can be **IMSI**.

Response Parameters

QueryBatchDevicesInfoOutDTO

Parameter	Description
$totalCount	Indicates the number of queried records.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records on each page.
$devices	Indicates the device pagination information list. For details, see **QuerySingleDeviceInfoOutDTO structure**.

QuerySingleDeviceInfoOutDTO structure

Parameter	Description
$deviceId	Identifies a device.
$gatewayId	Identifies a gateway.

Parameter	Description
$nodeType	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
$createTime	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$lastModifiedTime	Indicates the last time when the device is modified.
$deviceInfo	Indicates information about the device. For details, see **DeviceInfo structure**.
$services	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Description
$nodeId	Identifies a device.
$name	Indicates the device name.
$description	Indicates the device description.
$manufacturerId	Uniquely identifies a manufacturer.
$manufacturerName	Indicates the manufacturer name.
$mac	Indicates the MAC address of the device.
$location	Indicates the device location.
$deviceType	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
$model	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
$swVersion	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
$fwVersion	Indicates the firmware version of the device.
$hwVersion	Indicates the hardware version of the device.
$protocolType	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Parameter	Description
$bridgeId	Identifies the bridge through which the device accesses the IoT platform.
$status	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
$statusDetail	Indicates the device status. The specific value is determined by the value of **status**. For details, see **status and statusDetail**.
$mute	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
$supportedSecurity	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
$isSecurity	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
$signalStrength	Indicates the signal strength of the device.
$sigVersion	Indicates the SIG version of the device.
$serialNumber	Indicates the serial number of the device.
$batteryLevel	Indicates the battery level of the device.

status and statusDetail

$status	$statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Description
$serviceId	Identifies a service.
$serviceType	Indicates the service type.
$serviceInfo	Indicates service information of the device. For details, see **ServiceInfo structure**.
$data	Indicates an attribute value pair.
$eventTime	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Description
$muteCmds	Indicates the device command list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100218	The gatewayId and pageNo cannot be both null.	The **gatewayId** and **pageNo** parameters cannot be null at the same time. Recommended handling: Check whether **gatewayId** or **pageNo** is set.
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.8.3 Historical Device Data Query

Typical Scenario

The IoT platform receives and saves service data reported by devices during daily operation. The storage duration of device data can be configured by calling the API for modifying device information and the device data can be stored for a maximum of 90 days. If an NA needs to view the historical data reported by a device to the IoT platform, the NA can call this API to obtain the data.

API Function

This API is used by an NA to query historical data reported by a specified device to the IoT platform based on the device ID.

API Description

```
public function queryDeviceDataHistory($qddhInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qddhInDTO	Mandatory	query	For details, see **QuerySingleDeviceInfoOutDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceDataHistoryInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$deviceId	Mandatory	query	Identifies a device.
$gatewayId	Mandatory	query	Identifies a gateway.
$serviceId	Optional	query	Identifies a service.

Parameter	Mandatory or Optional	Location	Description
$property	Optional	query	Indicates the service attribute.
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the number of records on each page. The default value is **1**.
$startTime	Optional	query	Indicates the start time. Historical data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	query	Indicates the end time. Historical data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDataHistoryOutDTO

Parameter	Description
$totalCount	Indicates the number of queried records.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records on each page.
$deviceDataHistoryDTOs	Indicates a list of historical device data. For details, see **DeviceDataHistoryDTO structure**.

DeviceDataHistoryDTO structure

Parameter	Description
$serviceId	Identifies a service.

Parameter	Description
$deviceId	Identifies a device.
$gatewayId	Identifies a gateway.
$appId	Uniquely identifies an NA.
$data	Indicates the data reported by the device.
$timestamp	Indicates the timestamp when the data is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description. For example, check whether the value of **pageSize** exceeds 2000.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.8.4 Querying Historical Device Shadow Data

Typical Scenario

When an NA modifies the configuration of a device shadow by calling the API for modifying device shadow information, the IoT platform saves the modification record. If the NA needs to view historical configuration records of the device shadow, the NA can call this API to obtain the records.

API Function

This API is used by an NA to query historical configuration data about a device shadow based on the device ID.

API Description

```
public function queryDeviceDesiredHistory($qddhInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qddhInDTO	Mandatory	query	For details, see **QueryDeviceDesiredHistoryInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceDesiredHistoryInDTO

Parameter	Mandatory or Optional	Location	Description
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$deviceId	Mandatory	query	Identifies a device.
$gatewayId	Mandatory	query	Identifies a gateway.
$serviceId	Optional	query	Identifies a service.
$property	Optional	query	Indicates the service attribute.
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the number of records on each page. The default value is **1**.

Parameter	Mandatory or Optional	Location	Description
$startTime	Optional	query	Indicates the start time. Historical data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	query	Indicates the end time. Historical data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDesiredHistoryOutDTO

Parameter	Description
$totalCount	Indicates the number of queried records.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records on each page.
$DeviceDesiredHistoryDTO	Indicates a list of historical device data. For details, see **DeviceDesiredHistoryDTO structure**.

DeviceDesiredHistoryDTO structure

Parameter	Description
$serviceId	Identifies a service.
$deviceId	Identifies a device.
$gatewayId	Identifies a gateway.
$appId	Uniquely identifies an NA.
$desired	Indicates the data reported by the device.
$timestamp	Indicates the timestamp when the data is configured. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.8.5 Querying Service Capabilities of a Device

Typical Scenario

If an NA needs to know which service attributes can be reported by a device and which commands can be delivered to the device, the NA can call this API to query the device service capabilities defined in the profile file of the device on the IoT platform.

API Function

This API is used by an NA to query device service capabilities, such as service attributes and device commands.

API Description

```
public function queryDeviceCapabilities($qdcInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qdcInDTO	Mandatory	query	For details, see **QueryDeviceCapabilitiesInDTO structure**.

Parameter	Mandatory or Optional	Location	Description
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCapabilitiesInDTO

Parameter	Mandatory or Optional	Location	Description
$gatewayId	Optional	query	Identifies a gateway.
$appId	Mandatory	query	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$deviceId	Optional	query	Identifies a device.

Response Parameters

QueryDeviceCapabilitiesOutDTO

Parameter	Description
$deviceCapabilities	Indicates the query result list. For details, see **DeviceCapabilityDTO structure**.

DeviceCapabilityDTO structure

Parameter	Description
$deviceId	Identifies a device.
$serviceCapabilities	Indicates the service capability list. For details, see **ServiceCapabilityDTO structure**.

ServiceCapabilityDTO structure

Parameter	Description
$serviceId	Identifies a service.
$serviceType	Indicates the service type.
$option	Indicates a service option.
$description	Indicates the service description.
$commands	Indicates the supported commands. For details, see **ServiceCommand structure**.
$properties	Indicates the attribute list. For details, see **ServiceProperty structure**.

ServiceCommand structure

Parameter	Description
$commandName	Indicates the command name.
$paras	Indicates the attribute list. For details, see **ServiceCommandPara structure**.
$responses	Indicates the response list. For details, see **ServiceCommandResponse structure**.

ServiceCommandPara structure

Parameter	Description
$paraName	Indicates the parameter name.
$dataType	Indicates the data type.
$required	Indicates whether the parameter is mandatory.
$min	Indicates the minimum value of the attribute.
$max	Indicates the maximum value of the attribute.
$step	Indicates the step.
$maxLength	Indicates the maximum length.
$unit	Indicates the unit (symbol).
$enumList	Indicates the enumeration type list.

ServiceCommandResponse structure

Parameter	Description
$responseName	Indicates the response name.
$paras	Indicates the attribute list. For details, see **ServiceCommandPara structure**.

ServiceProperty structure

Parameter	Description
$propertyName	Indicates the attribute name.
$dataType	Indicates the data type.
$required	Indicates whether the parameter is mandatory.
$min	Indicates the minimum value of the attribute.
$max	Indicates the maximum value of the attribute.
$step	Indicates the step.
$maxLength	Indicates the maximum length.
$method	Indicates the access method. The values are as follows: ● **R**: Readable ● **W**: Writable ● **E**: Observable
$unit	Indicates the unit (symbol).
$enumList	Indicates the enumeration type list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9 Device Group Management

3.3.9.1 Creating a Device Group

Typical Scenario

An NA can call this API to create device groups on the IoT platform, and allocate devices to different device groups for group management. A device can be bound to multiple device groups.

When the NA needs to perform operations on devices (such as upgrading device software and firmware or delivering commands to devices in batches), the NA can select devices to be operated by device group.

API Function

This API is used by an NA to create device groups on the IoT platform to manage devices by group.

API Description

```
public function createDeviceGroup($cdgInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$cdgInDTO	Mandatory	body	For details, see **CreateDeviceGroupInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateDeviceGroupInDTO

Parameter	Mandatory or Optional	Location	Description
$name	Mandatory	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Optional	body	Indicates the device group description.
$appId	Optional	body	If the device belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Location	Description
$maxDevNum	Optional	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.
$deviceIds	Optional	body	Identifies the devices to be added to the device group.
id	Optional	body	Identifies a device group.

Response Parameters

StatusCode: 200 ok

CreateDeviceGroupOutDTO

Parameter	Description
$name	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Indicates the device group description.
$id	Identifies a device group. The device group ID is automatically generated by the IoT platform.
$appId	Uniquely identifies an NA.
$maxDevNum	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Indicates the current number of devices in the device group.
deviceIds	Identifies the devices to be added to the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100607	The devGroup has reached the limit.	The number of device groups reaches the limit. Recommended handling: Check whether the number of created device groups reaches the upper limit specified in the license.
400	100609	Too much devices to add.	Too many devices are added to the device group. Recommended handling: Ensure that the number of device IDs contained in **deviceIds** is within the range specified by **maxDevNum**.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.2 Deleting a Device Group

Typical Scenario

If a device group is no longer needed on the IoT platform due to group changes, an NA can call this API to delete a specified device group.

API Function

This API is used by an NA to delete the configuration information about a device group by device group ID on the IoT platform.

API Description

```
public function deleteDeviceGroup($devGroupId, $accessAppId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$devGroupId	Mandatory	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.

Parameter	Mandatory or Optional	Location	Description
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

StatusCode: 200 ok

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.3 Modifying a Device Group

Typical Scenario

If information about a device group (such as the device group name and the device quantity limit in the device group) needs to be modified due to service changes, an NA can call this API to modify the information.

API Function

This API is used to modify the information of a specified device group on the IoT platform.

API Description

```
public function modifyDeviceGroup($mdgInDTO, $devGroupId, $accessAppId,
$accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$mdgInDTO	Mandatory	body	For details, see **ModifyDeviceGroupInDTO structure**.
$devGroupId	Mandatory	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
$accessAppId	Mandatory	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Location	Description
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceGroupInDTO

Parameter	Mandatory or Optional	Location	Description
$name	Mandatory	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Optional	body	Indicates the device group description.
$maxDevNum	Optional	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.

Response Parameters

ModifyDeviceGroupOutDTO

Parameter	Description
$name	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Indicates the device group description.
$id	Identifies a device group.
$appId	Identifies the application to which the device group belongs.
$maxDevNum	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
$curDevNum	Indicates the current number of devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.4 Querying Details About a Device Group

Typical Scenario

An NA can call this API to query information of all the created device groups to check the group details and usage of the device groups.

API Function

This API is used by an NA to query information about all created device groups on the IoT platform.

API Description

```
public function queryDeviceGroups($qdgInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qdgInDTO	Mandatory	query	For details, see **QueryDeviceGroupsInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceGroupsInDTO

Parameter	Mandatory or Optional	Location	Description
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$pageNo	Optional	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the number of device group records on each page. The default value is **1**.
$name	Optional	query	Indicates the device group name.

Response Parameters

QueryDeviceGroupsOutDTO

Parameter	Description
$totalCount	Indicates the total number of device groups.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of device group records on each page.
$list	Indicates details about the device group. For details, see **QuerySingleDeviceGroupOutDTO structure**.

QuerySingleDeviceGroupOutDTO structure

Parameter	Description
$name	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Indicates the device group description.
$id	Identifies a device group.
$appId	Identifies the application to which the device group belongs.
$maxDevNum	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.

Parameter	Description
$curDevNum	Indicates the current number of devices in the device group.
$creator	Indicates the name of the user who created the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.5 Querying Information About a Specified Device Group

Typical Scenario

An NA can call this API to query information about a specified device group to check the usage of the device group.

API Function

This API is used by an NA to query the information about a device group by device group ID on the IoT platform.

API Description

```
public function querySingleDeviceGroup($devGroupId, $accessAppId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$devGroupId	Mandatory	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QuerySingleDeviceGroupOutDTO

Parameter	Description
$name	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
$description	Indicates the device group description.
$id	Identifies a device group.
$appId	Identifies the application to which the device group belongs.
$maxDevNum	Indicates the maximum number of devices in the device group.
$curDevNum	Indicates the current number of devices in the device group.

Parameter	Description
$creator	Indicates the name of the user who created the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.6 Querying Members in a Specified Device Group

Typical Scenario

An NA can call this API to query information about members in a specified device group.

API Function

This API is used by an NA to query the information about a device in a specified device group on the IoT platform.

API Description

```
public function queryDeviceGroupMembers($qdgmInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qdgmInDTO	Mandatory	query	For details, see **QueryDeviceGroupMembersInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceGroupMembersInDTO

Parameter	Mandatory or Optional	Location	Description
$devGroupId	Mandatory	query	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$pageNo	Optional	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.

Parameter	Mandatory or Optional	Location	Description
$pageSize	Optional	query	Indicates the number of devices on each page. The default value is **10**.

Response Parameters

QueryDeviceGroupMembersOutDTO

Parameter	Description
$totalCount	Indicates the number of devices in the device group.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of devices on each page.
$deviceIds	Identifies the devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	107001	The serviceId is not exist.	The service ID does not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.7 Adding Members to a Device Group

Typical Scenario

An NA can call this API to add a new device or an existing device to a specified device group. Before adding a device to a device group, you are advised to query the current number of devices and the maximum number of devices allowed in the device group by calling the API for querying information about a specified device group.

API Function

This API is used by an NA to add devices to a specified device group on the IoT platform.

API Description

```
public function addDevicesToGroup($dgwdlDTO, $accessAppId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$dgwdlDTO	Mandatory	body	For details, see **DeviceGroupWithDeviceListDTO structure**.
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeviceGroupWithDeviceListDTO

Parameter	Mandatory or Optional	Location	Description
$devGroupId	Mandatory	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
$deviceIds	Mandatory	body	Identifies the devices to be added to the device group.

Response Parameters

DeviceGroupWithDeviceListDTO

Parameter	Description
$devGroupId	Identifies a device group.
$deviceIds	Identifies the devices to be added to the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.9.8 Deleting Members from a Device Group

Typical Scenario

If one or more devices in a device group do not belong to the device group any longer, an NA can call this API to delete them from the device group.

API Function

This API is used by an NA to delete devices from a specified device group on the IoT platform.

API Description

```
public function deleteDevicesFromGroup($dgwdlDTO, $accessAppId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$dgwdlDTO	Mandatory	body	For details, see **DeviceGroupWithDeviceListDTO structure**.
$accessAppId	Optional	query	If the device group belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeviceGroupWithDeviceListDTO

Parameter	Mandatory or Optional	Location	Description
$devGroupId	Mandatory	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
$deviceIds	Mandatory	body	Identifies devices to be deleted from the device group.

Response Parameters

response:

Status Code: 200 OK

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

3.3.10 Device Upgrade

3.3.10.1 Querying a Version Package List

Typical Scenario

Before upgrading the device version, an NA can call this API to query the version upgrade packages that have been uploaded to the IoT platform to ensure that the target version package has been uploaded.

API Function

This API is used by an NA to query a list of uploaded version packages that meet a specified condition.

API Description

```
public function queryUpgradePackageList($quplInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$quplInDTO	Mandatory	query	For details, see **QueryUpgradePackageListInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradePackageListInDTO

Parameter	Mandatory or Optional	Location	Description
$fileType	Optional	query	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
$deviceType	Optional	query	Indicates the type of the device to which the version package is applicable.
$model	Optional	query	Indicates the model of the device to which the version package is applicable.
$manufacturerName	Optional	query	Indicates the manufacturer of the device to which the version package is applicable.
$version	Optional	query	Indicates the version of the version package.
$pageNo	Optional	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.

Parameter	Mandatory or Optional	Location	Description
$pageSize	Optional	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradePackageListOutDTO

Parameter	Description
$data	Indicates the version package list. For details, see **QueryUpgradePackageOutDTO structure**.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records to be displayed on each page.
$totalCount	Indicates the total number of query results.

QueryUpgradePackageOutDTO structure

Parameter	Description
$fileId	Identifies a version package.
$name	Indicates the version package name.
$version	Indicates the version of the version package.
$fileType	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
$deviceType	Indicates the type of the device to which the version package is applicable.
$model	Indicates the model of the device to which the version package is applicable.
$manufacturerName	Indicates the manufacturer of the device to which the version package is applicable.
$protocolType	Indicates the protocol used by the device to which the version package is applicable.
$description	Indicates the version package description.

Parameter	Description
$date	Indicates the date on which the version package was generated.
$uploadTime	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the values of **pageNo** and **pageSize** in the API request are within the valid ranges.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

3.3.10.2 Querying a Specified Version Package

Typical Scenario

Before upgrading the device version, an NA can call this API to query the target version upgrade package to ensure that it has been uploaded to the IoT platform.

API Function

This API is used by an NA to query a specified version package based on the version package ID on the IoT platform. The version package ID can be obtained by calling the API for querying a version package list.

API Description

```
public function queryUpgradePackage($fileId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$fileId	Mandatory	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QueryUpgradePackageOutDTO

Parameter	Description
$fileId	Identifies a version package.
$name	Indicates the version package name.
$version	Indicates the version of the version package.
$fileType	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
$deviceType	Indicates the type of the device to which the version package is applicable.
$model	Indicates the model of the device to which the version package is applicable.
$manufacturerName	Indicates the manufacturer of the device to which the version package is applicable.

Parameter	Description
$protocolType	Indicates the protocol used by the device to which the version package is applicable.
$description	Indicates the version package description.
$date	Indicates the date on which the version package was generated.
$uploadTime	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

3.3.10.3 Deleting a Specified Version Package

Typical Scenario

If a device version package is no longer needed, an NA can call this API to delete the version package from the IoT platform.

API Function

This API is used by an NA to delete a specified version package from the IoT platform based on the version package ID. The ID of the version package to be deleted can be obtained by calling the API for querying a version package list.

API Description

```
public function deleteUpgradePackage($fileId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$fileId	Mandatory	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

response:

Status Code: 200 OK

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

3.3.10.4 Creating a Software Upgrade Task

Typical Scenario

If the device software needs to be upgraded, an NA can call this API to create a software upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the software of multiple devices on the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Description

```
public function createSoftwareUpgradeTask($cutInDTO, $accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$cutInDTO	Mandatory	body	For details, see **CreateUpgradeTaskInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateUpgradeTaskInDTO

Parameter	Mandatory or Optional	Location	Description
$fileId	Mandatory	body	Identifies the target firmware version.
$targets	Mandatory	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.
$policy	Optional	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Location	Description
$deviceGroups	Optional	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
$deviceType	Optional	body	Indicates the device type. This parameter must be specified when a device group is specified.
$model	Optional	body	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Mandatory or Optional	Location	Description
$manufacturerName	Optional	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
$devices	Optional	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Location	Description
$executeType	Mandatory	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
$startTime	Optional	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Optional	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$retryType	Optional	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.

Parameter	Mandatory or Optional	Location	Description
$retryTimes	Optional	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO

Parameter	Description
$operationId	Identifies an operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.

HTTP Status Code	Error Code	Error Description	Remarks
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

3.3.10.5 Creating a Firmware Upgrade Task

Typical Scenario

If the device firmware needs to be upgraded, an NA can call this API to create a firmware upgrade task for multiple devices. Before the upgrade, ensure that the target version package

has been uploaded to the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the firmware of multiple devices on the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Description

```
public function createFirmwareUpgradeTask($cutInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$cutInDTO	Mandatory	body	For details, see **CreateUpgradeTaskInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateUpgradeTaskInDTO

Parameter	Mandatory or Optional	Location	Description
$fileId	Mandatory	body	Identifies the target firmware version.
$targets	Mandatory	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.
$policy	Optional	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Location	Description
$deviceGroups	Optional	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
$deviceType	Optional	body	Indicates the device type. This parameter must be specified when a device group is specified.
$model	Optional	body	Indicates the device model. This parameter must be specified when a device group is specified.
$manufacturerName	Optional	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
$devices	Optional	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Location	Description
$executeType	Mandatory	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
$startTime	Optional	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Location	Description
$endTime	Optional	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$retryType	Optional	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
$retryTimes	Optional	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO

Parameter	Description
$operationId	Identifies an operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.

HTTP Status Code	Error Code	Error Description	Remarks
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

3.3.10.6 Querying the Result of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, an NA can call this API to query details about the upgrade task, including the configuration and execution status.

API Function

This API is used by an NA to query details about a software or firmware upgrade task, including the configuration and execution status.

API Description

```
public function queryUpgradeTask($operationId, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$operationId	Mandatory	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.

Parameter	Mandatory or Optional	Location	Description
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QueryUpgradeTaskOutDTO

Parameter	Description
$operationId	Identifies an operation task.
$createTime	Indicates the time when the operation task was created.
$startTime	Indicates the time when the operation task was started.
$stopTime	Indicates the time when the operation task was stopped.
$operateType	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
$targets	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
$policy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.
$status	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
$staResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
$extendPara	Indicates an extended parameter of the operation task. For details, see **extendPara request parameter**.

OperateDevices structure

Parameter	Description
$deviceGroups	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
$deviceType	Indicates the device type. This parameter must be specified when a device group is specified.
$model	Indicates the device model. This parameter must be specified when a device group is specified.
$manufacturerName	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
$devices	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Description
$executeType	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
$startTime	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$retryType	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
$retryTimes	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Description
$total	Indicates the number of operated devices.
$wait	Indicates the number of devices waiting to be operated.
$processing	Indicates the number of devices that are being operated.
$success	Indicates the number of devices that are operated successfully.
$fail	Indicates the number of devices that fail to be operated.
$stop	Indicates the number of devices that stop being operated.
$timeout	Indicates the number of devices that fail to be operated due to a timeout.

extendPara request parameter when **operateType** is set to **softwareUpgrade** or **firmwareUpgrade**

Parameter	Description
$fileVersion	Identifies the target version.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

3.3.10.7 Querying Details About Subtasks of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, the upgrade of each device involved in the task is a subtask (the number of subtasks is the same as that of the devices involved in the task). An NA can call this API to query details about subtasks of the upgrade task to check their execution status.

API Function

This API is used by an NA to query upgrade status of each device involved in a software or firmware upgrade task.

API Description

```
public function queryUpgradeSubTask($qustInDTO, $operationId, $accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qustInDTO	Mandatory	query	For details, see **QueryUpgradeSubTaskInDTO structure**.
$operationId	Mandatory	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradeSubTaskInDTO

Parameter	Mandatory or Optional	Location	Description
$subOperationStatus	Optional	query	Indicates subtask status. If this parameter is not specified, execution details of all subtasks under the operation task are queried. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
$pageNo	Optional	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.

Parameter	Mandatory or Optional	Location	Description
$pageSize	Optional	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeSubTaskOutDTO structure

Parameter	Description
$data	Indicates the subtask list information. For details, see **SubOperationInfo**.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records to be displayed on each page.
$totalCount	Indicates the total number of query results.

SubOperationInfo structure

Parameter	Description
$subOperationId	Identifies a subtask.
$createTime	Indicates the time when the subtask was created.
$startTime	Indicates the time when the subtask was started.
$stopTime	Indicates the time when the subtask was stopped.
$operateType	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
$deviceId	Identifies a device on which the subtask is performed.
$status	Indicates the subtask status. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.

Parameter	Description
$detailInfo	Indicates detailed description of the subtask status. In case of a failure, the value of this parameter is the failure cause.
$extendInfo	Indicates extended information of the subtask. The value of this parameter varies depending on the operation type.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

3.3.10.8 Querying an Upgrade Task List

Typical Scenario

An NA can call this API to query the created upgrade tasks to view the detailed information and execution status of each upgrade task.

API Function

This API is used by an NA to query details about upgrade tasks that meet specified conditions.

API Description

```
public function queryUpgradeTaskList($qutlInDTO, $accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Location	Description
$qutlInDTO	Mandatory	query	For details, see **QueryUpgradeTaskListInDTO structure**.
$accessToken	Mandatory	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradeTaskListInDTO

Parameter	Mandatory or Optional	Location	Description
$operationType	Optional	query	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**

Parameter	Mandatory or Optional	Location	Description
$operationStatus	Optional	query	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
$deviceType	Optional	query	Indicates the device type for which the operation task is intended.
$model	Optional	query	Indicates the device model for which the operation task is intended.
$manufacturerName	Optional	query	Indicates the manufacturer of the device for which the operation task is intended.
$deviceId	Optional	query	Identifies the device for which the operation task is intended.
$pageNo	Optional	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
$pageSize	Optional	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeTaskListOutDTO

Parameter	Description
$data	Indicates the task list. For details, see **OperationInfo structure**.
$pageNo	Indicates the page number.
$pageSize	Indicates the number of records to be displayed on each page.
$totalCount	Indicates the total number of query results.

OperationInfo structure

Parameter	Description
$operationId	Identifies an operation task.
$createTime	Indicates the time when the operation task was created.
$startTime	Indicates the time when the operation task was started.
$stopTime	Indicates the time when the operation task was stopped.
$operateType	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
$targets	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
$policy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.
$status	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
$staResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
$extendPara	Indicates extended information of the operation task. The value of this parameter varies depending on the operation type.

OperateDevices structure

Parameter	Description
$deviceGroups	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
$deviceType	Indicates the device type. This parameter must be specified when a device group is specified.
$model	Indicates the device model. This parameter must be specified when a device group is specified.
$manufacturerName	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
$devices	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Description
$executeType	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
$startTime	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$endTime	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
$retryType	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
$retryTimes	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Description
$total	Indicates the number of operated devices.
$wait	Indicates the number of devices waiting to be operated.
$processing	Indicates the number of devices that are being operated.
$success	Indicates the number of devices that are operated successfully.
$fail	Indicates the number of devices that fail to be operated.
$stop	Indicates the number of devices that stop being operated.
$timeout	Indicates the number of devices that fail to be operated due to a timeout.

Error Code

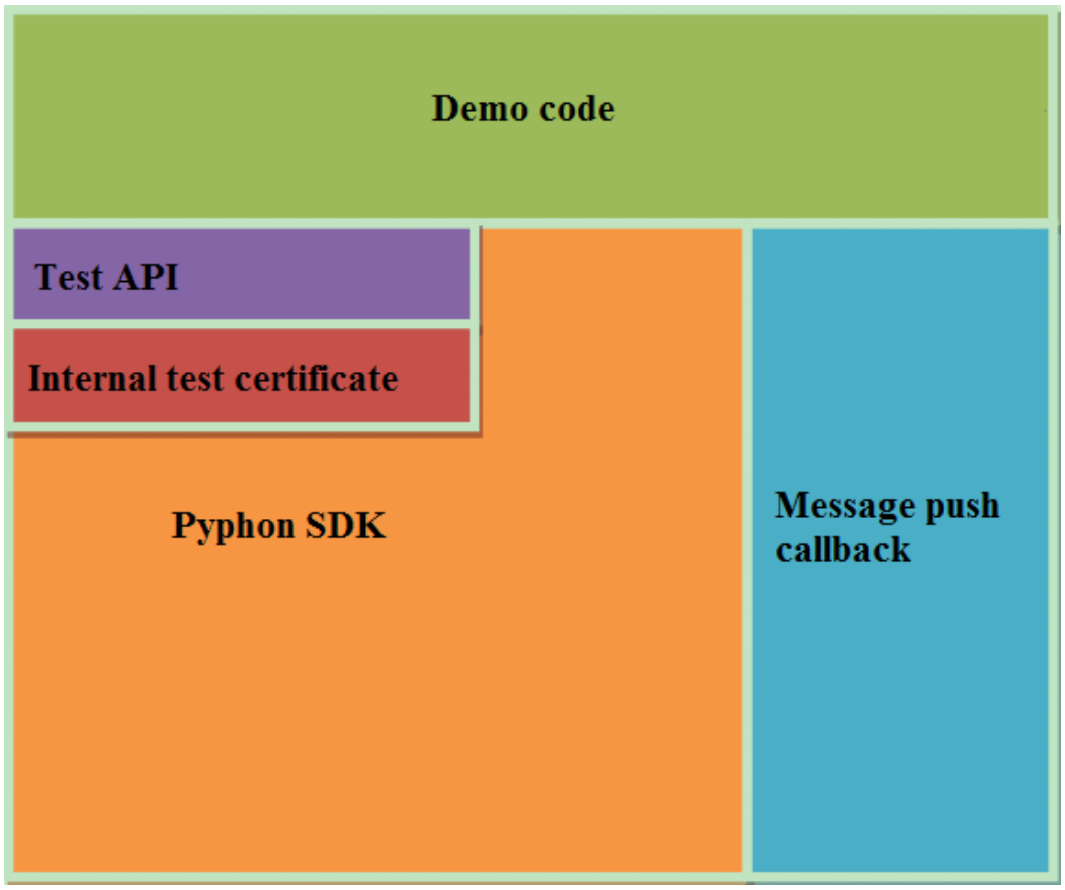
HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

4 Northbound Python SDK API Reference

4.1 SDK Demo Architecture and Usage Guide

- The demo code is the sample code used for calling SDK APIs, including initializing and calling each API. The demo code is for your reference only.
- The Python SDK uses Python methods to call northbound RESTful APIs provided by the IoT platform to communicate with the platform.
- The message push callback uses Python code to implement the callback API, which is called by the IoT platform to push messages to application servers. An application inherits the PushMessageReceiver class and rewrites the methods in the class to receive the content of a push message.
- The test API tests whether the basic functions between the SDK and IoT platform are available and generates test results. During the test, the internal test certificate and the certificate set by developers are used.



4.2 SDK Initialization Configuration and Test

4.2.1 Methods of the NorthApiClient Class

The methods of the NorthApiClient class are used to create an application instance. They are the prerequisites for calling other SDK APIs. The following describes the main methods:

Method	Description
def invokeAPI(httpMethod, url, headers, payload, clientInfo)	The API calling methods and the returned data vary according to the input parameters.

4.2.2 Methods of the ClientInfo Class

The methods of the ClientInfo class are used to set the basic information about the interconnection. The following describes the main methods:

Method	Description
platformIp	Indicates the IP address of the IoT platform.

Method	Description
platformPort	Indicates the port number of the IoT platform, for example, **8743**.
appId	Identifies an authorized application. This parameter is set to the ID allocated by the IoT platform for the partner's server.
accessToken	This parameter is set to the value of the access token obtained by calling the **Authentication** API

4.3 Service API List

4.3.1 Secure Application Access

After an NA obtains authentication information and connects to the IoT platform, the NA uses the authentication information to call other APIs.

4.3.1.1 Authentication

Typical Scenario

This API is called by an NA for access authentication when the NA accesses open APIs of the IoT platform for the first time. After the authentication of the NA expires, the NA must call this API to perform authentication again so that the NA can continue to access open APIs of the IoT platform.

API Function

This API is used by an NA to get authenticated before accessing open APIs of the IoT platform for the first time.

Note

The authentication API is the prerequisite for calling other APIs. Except the authentication API itself, other APIs must use the accessToken obtained by the authentication API.

If the access token is obtained for multiple times, the previous access token is invalid and the last access token obtained is valid. Do not obtain the access token through concurrent attempts.

API Description

```
def getAuthToken(self, clientInfo)
```

Class

Authentication

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
clientInfo	Mandatory	ClientInfo	-	For details, see **ClientInfo**.

ClientInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
platformIp	Mandatory	String(256)	path	Indicates the IP address of the IoT platform.
platformPort	Mandatory	String(256)	path	Indicates the port number of the IoT platform, for example, **8743**.
appId	Mandatory	String(256)	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
secret	Mandatory	String(256)	body	Indicates the password that corresponds to **appId**. It is used to log in to the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.

Return Value

AuthOutDTO structure

Parameter	Type	Description
scope	String(256)	Indicates the application permission scope, that is, the scope of IoT platform resources that can be accessed using the **accessToken**. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the type of the **accessToken**. This parameter has a fixed value of **bearer**.

Parameter	Type	Description
expiresIn	Number(256)	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.1.2 Refreshing a Token

Typical Scenario

The access token obtained by calling the Authentication API has a valid time. When the access token is about to expire, an NA can call this API to obtain a new access token.

API Function

This API is used by an NA to obtain a new access token from the IoT platform when the access token is about to expire.

API Description

```
def refreshAuthToken(self, arInDTO)
```

Class

Authentication

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
arInDTO	Mandatory	AuthRefreshInDTO	body	For details, see **AuthRefreshInDTO**.

AuthRefreshInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(256)	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
secret	Mandatory	String(256)	body	Indicates the password that corresponds to **appId**. It is used to log in to the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.

Parameter	Mandatory or Optional	Type	Location	Description
refreshToken	Mandatory	String(256)	body	Indicates the refresh token used for obtaining a new **accessToken**. The value of this parameter is obtained when the **Authentication** API is called.

Return Value

AuthRefreshOutDTO structure

Parameter	Type	Description
scope	String(256)	Indicates the applied permission range. This parameter has a fixed value of **default**.
tokenType	String(256)	Indicates the access token type. This parameter has a fixed value of **bearer**.
expiresIn	Number(256)	Indicates the validity time for the IoT platform to generate and return the **accessToken**, in seconds.
accessToken	String(256)	Indicates the authentication parameter that is used to access APIs of the IoT platform.
refreshToken	String(256)	Indicates the authentication parameter that is used to update the **accessToken**. The validity period of this parameter is 1 month.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100449	The device is freezed cant operate.	The user does not have the operation permission. Recommended handling: Check whether the user corresponding to appId has the permission to call the API.
400	102202	Required Parameter is null or empty.	Mandatory fields cannot be left blank. Recommended handling: Check whether the mandatory parameters in the request are set.

HTTP Status Code	Error Code	Error Description	Remarks
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100208	AppId or secret is not right.	**appId** or **secret** is incorrect. Recommended handling: ● Check whether **appId** and **secret** are correct. Specifically, check for new or missing characters. ● Check whether the IP address in the request path is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2 Device Management

An NA adds a device to the IoT platform and obtains the device ID and verification code. After connecting to the IoT platform, the device establishes a subordinate relationship with the NA.

4.3.2.1 Registering a Device (Verification Code Mode)

Typical Scenario

Before a device accesses the IoT platform by using verification code, an NA needs to call this API to register the device with the IoT platform and set a unique identification code of the device (such as the IMEI) as the verification code. Then, the device can use the unique identification code to get authenticated and connect to the IoT platform.

API Function

This API is used by an NA to register a device with the IoT platform. After registration, the device can connect to the IoT platform.

API Description

```
def regDirectDevice(self, rddInDto, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
rddInDto	Mandatory	RegDirectDeviceInDTO	body	For details, see RegDirectDeviceInDTO structure.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

RegDirectDeviceInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
customFields	Optional	List<**CustomField**>	Body	User-defined field list. Users can set customized fields.
deviceInfo	Optional	**DeviceInfoDTO**	Body	Specifies information about the device.
deviceInfo2	Optional	**DeviceInfo2**	Body	Indicates other device information.
deviceName	Optional	String(256)	Body	Specifies the name of a device.
endUserId	Optional	String(256)	Body	Identifies an end user. In the NB-IoT solution, this parameter is set to the IMSI of the device. In the Smart Home solution, this parameter is set to the application account.
groupId	Optional	String(256)	Body	Identifies the device group to which a device belongs.
imsi	Optional	String(1-64)	Body	Indicates the IMSI of an NB-IoT device.

Parameter	Mandatory or Optional	Type	Location	Description
isSecure	Optional	Boolean	Body	Indicates whether the device is secure. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure. **NOTE** If a user needs to register a secure device, this parameter must be specified.
location	Optional	**Location**	Body	Indicates the device location.
nodeId	Mandatory	String(256)	Body	Uniquely identifies a device. The value of this parameter must be the same as the device ID reported by the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
psk	Optional	String(8-32)	Body	If the pre-shared key (PSK) is specified in the request, the IoT platform uses the specified PSK. If the PSK is not specified in the request, the PSK is generated by the IoT platform. The value is a string of characters, including upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.
tags	Optional	List<**TagDTO2**>	Body	Indicates the tag of a device.

Parameter	Mandatory or Optional	Type	Location	Description
timeout	Optional	Integer(>=0)	Body	Indicates the validity period for device registration. When this API is called to register a device, the device can be bound within the validity period. If the device is not bound within the validity period, the registration information will be deleted. The value ranges from 0 to 2147483647. If this parameter is set to **0**, the device verification code is always valid. (The recommended value is **0**.) The default value is **180**. The default value can be configured. For details, contact the IoT platform maintenance personnel. Unit: second
verifyCode	Conditionally optional Mandatory	String(256)	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. In the NB-IoT solution, this parameter is mandatory and must be set to the same value as **nodeId**.
productId	Optional	String(256)	Body	Identifies the product to which the device belongs.
account	Optional	String	Body	Indicates the IPC device account.

CustomField structure

Parameter	Mandatory or Optional	Type	Location	Description
fieldName	Optional	String(256)	Body	Indicates the field name.
fieldType	Optional	String(256)	Body	Indicates the field type.

Parameter	Mandatory or Optional	Type	Location	Description
fieldValue	Optional	String(256)	Body	Indicates the field value.

DeviceInfoDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Optional	String(256)	Body	Identifies a device.
name	Optional	String(256)	Body	Specifies the name of a device.
description	Optional	String(2048)	Body	Indicates the device description.
manufacturerId	Optional	String(256)	Body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	Body	Indicates the manufacturer name.
mac	Optional	String(256)	Body	Indicates the MAC address of the device.
location	Optional	String(2048)	Body	Indicates the device location.
deviceType	Optional	String(256)	Body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Mandatory	String(256)	Body	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	Optional	String(256)	Body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	Body	Indicates the firmware version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
hwVersion	Optional	String(256)	Body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	Body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	Optional	String(256)	Body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	Body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	Optional	String(256)	Body	Indicates the device status details, which vary according to the value of **status**. ● When **status** is **ONLINE**, the value options are **NONE**, **COMMUNICATION_ERROR**, **CONFIGURATION_ERROR**, **BRIDGE_OFFLINE**, **FIRMWARE_UPDATING**, **DUTY_CYCLE**, and **NOT_ACTIVE**. ● When **status** is **OFFLINE**, the value options are **NONE** and **CONFIGURATION_PENDING**.
mute	Optional	String	Body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	Body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.

Parameter	Mandatory or Optional	Type	Location	Description
isSecurity	Optional	String	Body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	Body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	Body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	Body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	Body	Indicates the battery level of the device.

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceInfo2 structure

Parameter	Mandatory or Optional	Type	Location	Description
region	Optional	String(256)	Body	Indicates the region where the device is located.
timezone	Optional	String(256)	Body	Specifies the time zone where a device is located.
activeServiceCount	Optional	Integer	Body	Indicates the number of active services of the device.

Location structure

Parameter	Mandatory or Optional	Type	Location	Description
accuracy	Optional	Double	Body	Indicates the accuracy. The minimum value is **0**.

Parameter	Mandatory or Optional	Type	Location	Description
crashInformation	Optional	String	Body	Indicates the emergency information.
description	Optional	String	Body	Indicates the location description.
heading	Optional	String	Body	Indicates the direction information.
language	Optional	String	Body	Indicates the language.
latitude	Optional	Double	Body	Indicates the latitude of the device. The value ranges from -90 to 90.
longitude	Optional	Double	Body	Indicates the longitude of the device. The value ranges from -180 to 180.
numberOfPassengers	Optional	String	Body	Indicates the number of passengers.
region	Optional	String	Body	Indicates the region where the device is located.
time	Optional	DateTime	Body	Indicates the time information.
vehicleSpeed	Optional	String	Body	Indicates the vehicle speed.

TagDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.

Return Value

RegDirectDeviceOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device.
verifyCode	String(256)	Indicates a verification code that can be used by the device to obtain the device ID and password. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
timeout	Number	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.
psk	String(32)	Indicates a random PSK. If the PSK is carried in the request, it is used. Otherwise, the IoT platform generates a random PSK.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	103028	The license pool resources.	The license resources have been used up.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100416	The device has already been bounded.	The device has been bound to the IoT platform. Recommended handling: Check whether the device is registered.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application has not been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
403	600002	The product not existed.	The product does not exist. Recommended handling: Check whether **productId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100412	The amount of device has reached the limit.	The number of devices under the current application reaches the upper limit. Recommended handling: Check whether the number of devices under the current application reaches the upper limit.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	103026	The license is not exist.	The license does not exist. Recommended handling: An internal license error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.2 Refreshing a Device Key

Typical Scenario

If the unique identification code of a device that has been registered with the IoT platform changes (for example, a device is replaced), an NA needs to call this API to update the unique identification code of the device and rebind the device.

NOTE

The device password can be updated only when the device is offline.

API Function

This API is used by an NA to update the node ID of a device that has been registered with the IoT platform, and rebind the device with the device ID unchanged.

API Description

```
def refreshDeviceKey(self, rdkInDTO, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
rdkInDTO	Mandatory	RefreshDeviceKeyInDTO structure	body	For details, see RefreshDeviceKeyInDTO structure.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

RefreshDeviceKeyInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(256)	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.

Parameter	Mandatory or Optional	Type	Location	Description
verifyCode	Optional	String(256)	body	Indicates the verification code of the device. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform. You are advised to set this parameter to the value of **nodeId**.
nodeId	Optional	String(256)	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. ● If the value is **null**, the value of this parameter remains unchanged. ● If the value is not **null**, the value of this parameter is updated. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
timeout	Optional	Number	body	Indicates the validity period of the verification code, in units of seconds. The value is an integer greater than or equal to **0**. ● If this parameter is set to **null**, the default value **180** prevails. ● If this parameter is set to **0**, the verification code never expires. ● If this parameter is not set to **0**, the verification code expires after the specified time elapses.

Return Value

RefreshDeviceKeyOutDTO structure

Parameter	Type	Description
verifyCode	String(256)	Indicates a verification code that can be used by the device to obtain the device ID and password. If this parameter is specified in the request, it is returned in the response. If this parameter is not specified in the request, it is automatically generated by the IoT platform.
timeout	Number	Indicates the validity period of the verification code, in seconds. The device must connect to the IoT platform within this period.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
400	100003	Invalid verify code.	The verification code is invalid. Recommended handling: Check whether **verifyCode** carried in the API request is correct. If **verifyCode** is not carried in the API request, contact IoT platform maintenance personnel.
400	100007	Bad request message.	The request message contains invalid parameters. Recommended handling: The value of **deviceId** is not assigned. Set this parameter based on the description of request parameters.
400	100426	The nodeId is duplicated.	The value of **nodeId** is duplicated. Recommended handling: Check whether **nodeId** carried in the API request is correct.
400	100610	Device is not active.	The device has not been activated. Recommended handling: Check whether the device has been connected to the IoT platform and activated.

HTTP Status Code	Error Code	Error Description	Remarks
400	100611	Device is online.	The device is online. Recommended handling: Enable the device to go offline or disconnect the device from the IoT platform.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100025	AppId for auth not exist.	The application ID used for authentication does not exist. Recommended handling: ● Check whether the value is assigned to **app_key** in the header of the request structure. ● If this API is called using HTTP, contact IoT platform maintenance personnel to check whether the name of **appId** in the header is **app_key** or **x-app-key**.
403	100203	The application is not existed.	The authorized application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.3 Modifying Device Information

Typical Scenario

After an NA registers a device with the IoT platform and the basic information about the device changes, the NA can also call this API to modify device information on the IoT platform.

API Function

This API is used to modify the basic information about a device, including the device type, device model, manufacturer, and access protocol.

API Description

```
def modifyDeviceInfo(self, mdiInDto, deviceId, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
mdiInDto	Mandatory	ModifyDeviceInfoInDTO	body	For details, see ModifyDeviceInfoInDTO structure.
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceInfoInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
customFields	Optional	List<**CustomField**>	Body	User-defined field list. Users can set customized fields.
deviceConfig	Optional	**DeviceConfigDTO**	body	Device configuration information

Parameter	Mandatory or Optional	Type	Location	Description
deviceType	Optional	String(1~256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**. After registering a device, you must change the device type to be the same as that defined in the profile file.
endUser	Optional	String(1~256)	body	Indicates an end user. If the device is a directly connected device, this parameter is optional. If it is a non-directly connected device, this parameter can be set to **null**.
location	Optional	String(1~1024)	body	Indicates the device location.
manufacturerId	Optional	String(1~256)	body	Uniquely identifies a manufacturer. After registering a device, you must change the manufacturer ID to be the same as that defined in the profile file.
manufacturerName	Optional	String(1~256)	body	Indicates the manufacturer name.
model	Optional	String(1~256)	body	Indicates the device model, which is defined by the manufacturer. After registering a device, you must change the device model to be the same as that defined in the profile file.
mute	Optional	Enum	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
name	Optional	String(1~256)	body	Indicates the device name.
organization	Optional	String(1~256)	body	Indicates the organization to which the device belongs.

Parameter	Mandatory or Optional	Type	Location	Description
protocolType	Optional	String(1~256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**. After registering a device, you must change the protocol type to be the same as that defined in the profile file.
region	Optional	String(1~256)	body	Indicates the region information about a device.
timezone	Optional	String(1~256)	body	Indicates the time zone where the device is located. The time zone code is used. For example, the time zone code of Shanghai time zone is **Asia/Shanghai**.
imsi	Optional	String(1-64)	Body	Indicates the IMSI of an NB-IoT device.
ip	Optional	String(128)	Body	Indicates the device IP address.
isSecure	Optional	Boolean	body	Indicates the security status of the device. The default value is **false**. ● **true**: The device is secure. ● **false**: The device is not secure.
psk	Optional	String(8~32)	body	Indicates the PSK. The value is a string of characters that consist of only upper-case letters A to F, lower-case letters a to f, and digits 0 to 9.
tags	Optional	List<**Tag2**>	Body	Indicates the tag of a device.

CustomField structure

Parameter	Mandatory or Optional	Type	Location	Description
fieldName	Optional	String(256)	Body	Indicates the field name.
fieldType	Optional	String(256)	Body	Indicates the field type.

Parameter	Mandatory or Optional	Type	Location	Description
fieldValue	Optional	String(256)	Body	Indicates the field value.

DeviceConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataConfig	Optional	DataConfigDTO	Indicates the data configuration information.

DataConfigDTO structure

Parameter	Mandatory or Optional	Type	Description
dataAgingTime	Optional	Integer	Indicates the data aging time. The value range is 0-90. Unit: day.

Tag2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.
tagType	Optional	Integer	body	Indicates the tag type.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100440	The isSecure is invalid.	The value of **isSecure** is incorrect.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	500004	The amount of frozen devices has reached the limit.	The number of frozen devices has reached the upper limit.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	100441	The amount of nonSecure device has reached the limit.	The number of non-security devices has reached the upper limit.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.4 Deleting a Device (Verification Code Mode)

Typical Scenario

If a device that has been registered with the IoT platform does not need to connect to the platform, an NA can call this API to delete the device. If the device needs to connect to the IoT platform again, the NA must register the device again.

API Function

This API is used by an NA to delete a registered device from the IoT platform.

API Description

```
def deleteDirectDevice(self, deviceId, cascade, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
cascade	Optional	String	query	This parameter is valid only when the device is connected to a non-directly connected device. If this parameter is not set, the value **None** is used. ● **true**: The directly connected device and the non-directly-connected devices connected to it are deleted. ● **false**: The directly connected device is deleted but the non-directly-connected devices connected to it are not deleted. In addition, the attribute of the non-directly-connected devices is changed to directly-connected.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.5 Querying Device Activation Status

Typical Scenario

After an NA registers a device on the IoT platform, the activation status of the device is **false** before the device connects to the IoT platform for the first time. When the device connects to the IoT platform for the first time, the activation status of the device is **true** regardless of whether the device is online, offline, or abnormal. The NA can call this API to query the activation status of the device to check whether the device has connected to the IoT platform.

API Function

This API is used by an NA to query the activation status of a device on the IoT platform to determine whether the device has connected to the IoT platform.

API Description

```
def queryDeviceStatus(self, deviceId, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

QueryDeviceStatusOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device.
activated	Boolean	Indicates whether the device is activated through a verification code. ● **true**: The device is activated. ● **false**: The device is not activated.
name	String(256)	Specifies the name of a device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.6 Querying Device Shadow Information

Typical Scenario

When a device is in the offline or abnormal state, an NA cannot deliver configuration information to the device by sending a command. In this case, the NA can deliver the configuration information to the device shadow. When the device goes online, the device shadow will deliver the configuration information to the device. The NA can call this API to check the device configuration information and the latest data reported by the device on the device shadow.

API Function

This API is used by an NA to query the device shadow information of a device, including the device configuration information (in the desired section) and the latest data reported by the device (in the reported section).

API Description

```
def queryDeviceShadow(self, deviceId, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

QueryDeviceShadowOutDTO structure

Parameter	Type	Description
deviceId	String(36)	Uniquely identifies a device.
gatewayId	String(36)	Identifies a gateway.
nodeType	Enum	Indicates the device type.
createTime	String(256)	Indicates the device creation time.
lastModifiedTime	String(256)	Indicates the last modification time.
deviceInfo	DeviceInfo	Indicates the device information. For details, see **DeviceInfo structure**.
services	List<DeviceServiceB>	Indicates the device service capabilities. For details, see **DeviceServiceB structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Uniquely identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer. Set this parameter to the value defined in the profile file of the device.
manufacturerName	String(256)	Indicates the manufacturer name. Set this parameter to the value defined in the profile file of the device.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used. Set this parameter to the value defined in the profile file of the device, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model. Set this parameter to the value defined in the profile file of the device. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.

Parameter	Type	Description
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. Set this parameter to the value defined in the profile file of the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	String(256)	Indicates the device status. The specific value is determined by the value of **status**. For details, see status and statusDetail.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceServiceB structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
reportedProps	ObjectNode	Indicates the information reported by the device.
desiredProps	ObjectNode	Indicates the information delivered to the device.
eventTime	String(256)	Indicates the time when an event occurs.
serviceType	String(256)	Indicates the service type.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.2.7 Modifying Device Shadow Information

Typical Scenario

The IoT platform supports the creation of device shadows. Device shadows store the latest service attribute data reported by devices and service attribute configurations delivered by NAs. (Service attributes are defined in the device profile file.) If the device is offline or abnormal, the NA cannot deliver configuration to the device by issuing commands. In this

case, the NA can set the configuration to be delivered to the device shadow. When the device goes online again, the device shadow delivers the configuration to the device. The NA can call this API to modify the configuration information to be delivered to the device on the device shadow.

Each device has only one device shadow, which contains desired and report sections.

- The desired section stores the configurations of device service attributes. If a device is online, the configurations in the desired section are delivered to the device immediately. Otherwise, the configurations in the desired section are delivered to the device when the device goes online.
- The report section stores the latest service attribute data reported by devices. When a device reports data, the IoT platform synchronizes the data to the report section of the device shadow.

API Function

This API is used to modify the configuration information in the desired section of the device shadow. When the device goes online, the configuration information will be delivered to the device.

API Description

```
def modifyDeviceShadow(self, mdsInDTO, deviceId, appId, accessToken)
```

Class

DeviceManagement

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
mdsInDTO	Mandatory	ModifyDeviceShadowInDTO	body	For details, see ModifyDeviceShadowInDTO structure.
deviceId	Mandatory	String(256)	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceShadowInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceDesireds	Mandatory	List<ServiceDesiredDTO>	body	Indicates the configuration or status to be modified. For details, see ServiceDesiredDTO structure.

ServiceDesiredDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Optional	String(1-256)	body	Identifies a service.
desired	Optional	Object	body	Indicates the device status.

Return Value

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100425	The special deviceCapability is not exist.	The device template does not exist. Recommended handling: Check whether the device template has been uploaded to the IoT platform.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.

HTTP Status Code	Error Code	Error Description	Remarks
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100443	The property is forbidden to write.	The device attributes cannot be written.
403	10100 09	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	10100 05	pp_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.3 Batch Processing

NAs can perform batch operations on devices connected to the IoT platform through the batch processing API.

4.3.3.1 Creating a Batch Task

Typical Scenario

When an NA needs to perform an operation on a batch of devices, the NA can call this API to create a batch task. Currently, the supported batch operations include delivering pending commands to devices in batches.

API Function

This API is used by an NA to create a batch task for devices on the IoT platform.

API Description

```
def createBatchTask(self, btcInDTO, accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
btcInDTO	Mandatory	BatchTaskCreateInDTO	body	For details, see BatchTaskCreateInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

BatchTaskCreateInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String(64)	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform. The value of this parameter is obtained when the NA is created on the SP portal of the IoT platform.
param	Mandatory	**Object Node**	body	Indicates the task parameter, which varies depending on the value of **taskType**.
taskName	Mandatory	String	body	Indicates the task name. The value is a string of a maximum of 256 characters.
taskType	Mandatory	String	body	Indicates the task type. The values include **DeviceReg**, **DeviceCmd**, **DeviceLocation**, **DeviceMod**, **DeviceDel**, and **DeviceLicense**.
timeout	Mandatory	Integer	body	Indicates the timeout duration of a task, in minutes. The value range is 10 - 2880.
tags	Optional	List<**TagDTO2**>	body	Indicates the tag list.

ObjectNode structure

Parameter	Mandatory or Optional	Type	Location	Description
type	mandatory	String	body	Indicates the batch command type. This parameter is mandatory when **taskType** is set to **DeviceCmd**. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	Conditionally mandatory	List<String>	body	Indicates the device ID list. This parameter is mandatory when **type** is set to **DeviceList**.
deviceType	Conditionally mandatory	String	body	Indicates the device type. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
manufacturerId	Conditionally optional	String	body	Identifies the manufacturer. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
model	Conditionally optional	String	body	Indicates the device model. This parameter is mandatory when **type** is set to **DeviceType**. Set this parameter to the value defined in the profile file.
deviceLocation	Conditionally mandatory	String	body	Indicates the location of the device. This parameter is mandatory when **type** is set to **DeviceArea**.
groupList	Conditionally mandatory	List<String>	body	Indicates the group ID list or device group name list. When **type** is set to **GroupIdList**, set this parameter to the group ID. When **type** is set to **GroupList**, set this parameter to the device group name.
command	mandatory	**CommandDTO**	body	Indicates the command information.
callbackUrl	Conditionally optional	String	body	Indicates the push address of the command execution result.

Parameter	Mandatory or Optional	Type	Location	Description
maxRetransmit	Conditionally optional	Integer(0~3)	body	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	Conditionally optional	String	body	Indicates the group tag.

CommandDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	Mandatory	String(1-128)	body	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	Optional	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Mandatory or Optional	Type	Location	Description
tagName	Mandatory	String(1-128)	body	Indicates the tag name.

Parameter	Mandatory or Optional	Type	Location	Description
tagValue	Mandatory	String(1-1024)	body	Indicates the tag value.

Return Value

BatchTaskCreateOutDTO structure

Parameter	Type	Description
taskID	String	Identifies a batch task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	101001	Resource doesn't exist.	The resource does not exist.
200	105001	The batchTask count has reached the limit.	If the number of unfinished tasks is greater than or equal to 10, a message is returned, indicating that the number of tasks reaches the limit.
200	105002	The batchTask name has exist.	The task name exists. Recommended handling: Change the task name.
400	105201	The tagName and tagValue has been used on the platform.	The tagName and tagValue have been used on the IoT platform.

HTTP Status Code	Error Code	Error Description	Remarks
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
401	100028	The user has no right.	The user has no operation permission.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105202	The tag is not existed.	The tag does not exist.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.3.2 Querying Information About a Specified Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about the batch task, including the task status and the subtask completion status.

API Function

This API is used by an NA to query the information about a single batch task by task ID.

API Description

```
def queryOneTask(self, taskId, select, appId, accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
taskId	Mandatory	String	path	Identifies a batch task. The value of this parameter is obtained after the batch task is created.
select	Mandatory	String	query	Indicates an optional return value. The value can be **tag**. If this parameter is not specified, the value can be **None**.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Return Value

QueryOneTaskOutDTO structure

Parameter	Type	Description
appId	String	Indicates the **appId** to which the batch task belongs.
taskId	String	Identifies a batch task.
taskName	String	Indicates the name of the batch task.
operator	String	Indicates the operator who delivers the batch task.
taskFrom	String	Indicates the source of the batch task. ● **Portal**: The task is created on the SP portal. ● **Northbound**: The task is created by calling a northbound API.
taskType	String	Indicates the type of the batch task. The values include **DeviceReg**, **DeviceCmd**, **DeviceLocation**, **DeviceMod**, **DeviceDel**, and **DeviceLicense**.
status	String	Indicates the status of the batch task. The value options are **Pending**, **Running**, **Complete**, and **Timeout**.
startTime	String	Indicates the time when the batch task is created.
timeout	Integer	Indicates the time when the batch task times out, in seconds.
progress	Integer	Indicates the progress of the batch task. The value is a permillage ranging from 0 to 1000 and is rounded down.
totalCnt	Integer	Indicates the total number of tasks.
successCnt	Integer	Indicates the number of tasks that are successfully executed.
failCnt	Integer	Indicates the number of tasks that fail to be executed.
timeoutCnt	Integer	Indicates the number of tasks with execution timeout.
expiredCnt	Integer	Indicates the number of expired tasks that are not executed.

Parameter	Type	Description
completeCnt	Integer	Indicates the number of completed tasks, including successful, failed, and timed out tasks.
successRate	Integer	Indicates the task success rate. The value is a permillage ranging from 0 to 1000 and is rounded down.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.
tags	List<**TagDTO2**>	Indicates the tag list of the batch task.

ObjectNode structure

Parameter	Type	Description
type	String	Indicates the batch command type. The options include **DeviceList**, **DeviceType**, **DeviceArea**, **GroupList**, **Broadcast**, and **GroupIdList**.
deviceList	List<String>	Indicates the device ID list. The value is that returned when **type** is set to **DeviceList**.
deviceType	String	Indicates the type of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
manufacturerId	String	Identifies a manufacturer. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
model	String	Indicates the model of the device. The value is that returned when **type** is set to **DeviceType**. The value must be the same as that defined in the profile file.
deviceLocation	String	Indicates the location of the device. The value is that returned when **type** is set to **DeviceArea**.
groupList	List<String>	Indicates the group name list. The value is that returned when **type** is set to **GroupList**.
command	**CommandDTO**	Indicates the command information. .
callbackUrl	String	Indicates the push address of the command execution result..
maxRetransmit	Integer(0~3)	Indicates the maximum number of retransmissions of a command. The value ranges from 0 to 3.
groupTag	String	Indicates the group tag.

CommandDTO structure

Parameter	Type	Indicates the location description information.
serviceId	String(1-64)	Identifies the service corresponding to the command. The value of this parameter must be the same as the value of **serviceId** defined in the profile file.
method	String(1-128)	Indicates the name of a specific command under the service. The value of this parameter must be the same as the command name defined in the profile file.
paras	ObjectNode	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device.

TagDTO2 structure

Parameter	Type	Description
tagName	String(1-128)	Indicates the tag name.
tagValue	String(1-1024)	Indicates the tag value.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100019	Illegal request.	Invalid request. Recommended handling: Check whether the mandatory parameters in the request are set.
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

4.3.3.3 Querying Information About a Subtask of a Batch Task

Typical Scenario

After creating a batch task for devices, an NA can call this API to query information about a subtask of the batch task, including the subtask execution status and subtask content.

API Function

This API is used by an NA to query information about a subtask of a batch task based on specified conditions. This API applies to the current application.

API Description

```
def queryTaskDetails(self, qtdInDTO, accessToken)
```

Class

BatchProcess

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qtdInDTO	Mandatory	QueryTaskDetailsInDTO	query	For details, see QueryTaskDetailsInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryTaskDetailsInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	If the task belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
taskId	Mandatory	String	query	Identifies a batch task.
status	Optional	String	query	Indicates the task status. The value options are **Pending**, **Success**, **Fail**, and **Timeout**.
index	Optional	Integer	query	Indicates the index in a file. This parameter is used for querying details about a batch device registration task.
nodeId	Optional	String	query	Uniquely identifies the device. This parameter is used for querying details about a batch device registration task.

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Optional	String	query	Identifies a device. This parameter is used for querying details about a batch command delivery task.
commandId	Optional	String	query	Identifies a command. This parameter is used for querying details about a batch command delivery task.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.

Return Value

QueryTaskDetailsOutDTO structure

Parameter	Type	Description
pageNo	Long	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Long	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **1**.
totalCount	Long	Indicates the total number of records.
taskDetails	List<**QueryTaskDetailDTOCloud2NA**>	Indicates the task details list.

QueryTaskDetailDTOCloud2NA structure

Parameter	Type	Description
status	String	Indicates the task status. The value options are **Pending**, **WaitResult**, **Success**, **Fail**, and **Timeout**.
output	String	Indicates the output of a batch command delivery task.
error	String	Indicates the cause of error, in the format of {\"error_code\": \"****\", \"error_desc\":\"***** \"}.
param	**ObjectNode**	Indicates the task parameter, which varies depending on the value of **taskType**.

ObjectNode structure

Parameter	Type	Description
deviceId	String	Uniquely identifies a device. The value of this parameter is allocated by the IoT platform during device registration..
commandId	String	Uniquely identifies a device command. The value of this parameter is allocated by the IoT platform during command delivery.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	105005	The batchTask is not existed.	The batch task does not exist. Recommended handling: Check whether **taskId** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.4 Subscription Management

The IoT platform allows NAs to subscribe to device data. If the subscribed device data changes, the IoT platform pushes change notifications to NAs. The subscription management API must be used together with the **Message Push** API.

4.3.4.1 Subscribing to Service Data of the IoT Platform

Typical Scenario

An NA can subscribe to service data of a device on the IoT platform. When the service data changes (for example, the device is registered, the device reports data and the device status changes), the IoT platform can push change notifications to the NA. The NA can call this API to subscribe to different types of service change notifications.

API Function

This API is used by an NA to subscribe to service change notifications on the IoT platform. When the device status or data changes, the IoT platform pushes notifications to the NA.

API Description

```
def subDeviceBusinessData(self, sdbdInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
sdbdInDTO	Mandatory	SubDeviceBusinessDataInDTO	body	For details, see SubDeviceBusinessDataInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

SubDeviceBusinessDataInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
ownerFlag	Optional	String(256)	query	Indicates the owner flag of the callback URL. ● **false**: The callback URL owner is an authorizing application. ● **true**: The callback URL owner is an authorized application.
appId	Optional	String(256)	body	Indicates the application ID of the device.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type based on which an NA can receive notification messages pushed by the IoT platform. The value options are as follows: ● **bindDevice**: device binding; sending such a notification after subscription ● **deviceAdded**: device addition; sending such a notification after subscription ● **deviceInfoChanged**: device information change; sending such a notification after subscription ● **deviceDataChanged**: device data change; sending such a notification after subscription ● **deviceDatasChanged**: batch device data change; sending such a notification after subscription ● **deviceDeleted**: device deletion; sending such a notification after subscription ● **messageConfirm**: message confirmation; sending such a notification after subscription ● **commandRsp**: command response; sending such a notification after subscription ● **deviceEvent**: device event; sending such a notification after subscription ● **serviceInfoChanged**: service information change; sending such a notification after subscription ● **deviceModelAdded**: device model addition; sending such a notification after subscription ● **deviceModelDeleted**: device model deletion; sending such a notification after subscription ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change; sending such a notification after subscription

Parameter	Mandatory or Optional	Type	Location	Description
callbackUrl	Mandatory	String(1024)	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.
channel	Optional	String(32)	Body	Indicates the transmission channel. For the MQTT client, the value is **MQTT**. In other cases, the value is **HTTP**.

Response Parameters

SubscriptionDTO structure

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	The request callbackurl is illegal.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
409	100227	The resource is conflicted.	A resource conflict occurs. The notification type has been subscribed to. Recommended handling: Check whether the notification type has been subscribed.

4.3.4.2 Subscribing to Management Data of the IoT Platform

Typical Scenario

An NA can subscribe to management data of a device on the IoT platform. When operations are performed on the device (for example, device upgrade), the IoT platform notifies the NA of the operation status or results. The NA can call this API to subscribe to different types of device upgrade notifications on the IoT platform.

API Function

This API is used by an NA to subscribe to device upgrade notifications on the IoT platform. When the device is upgraded, the IoT platform sends a notification to the NA.

API Description

```
def subDeviceManagementData(self, sdmdInDTO, accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
smdInDTO	Mandatory	SubDeviceManagementDataInDTO	body	For details, see SubDeviceManagementDataInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

SubDeviceManagementDataInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. ● **swUpgradeStateChangeNotify**: software upgrade status change notification; sending such a notification after subscription ● **swUpgradeResultNotify**: software upgrade result notification; sending such a notification after subscription ● **fwUpgradeStateChangeNotify**: hardware upgrade status change notification; sending such a notification after subscription ● **fwUpgradeResultNotify**: hardware upgrade result notification; sending such a notification after subscription
callbackurl	Mandatory	String	body	Indicates the callback URL of a subscription, which is used to receive notification messages of the corresponding type. This URL must be an HTTPS channel callback URL and contain its port number. An example value is **https://XXX.XXX.XXX.XXX:443/callbackurltest**. **NOTE** The HTTP channel can be used only for commissioning.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100222	Internal server error.	The callback URL is invalid. Recommended handling: Check whether the callback URL in the request body is correct.
400	100228	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100229	Get AppKey from header failed.	Failed to obtain the AppKey from the message header.
500	100244	register out route fail.	Failed to register the route. Recommendation: Contact the IoT platform maintenance personnel.

4.3.4.3 Querying a Subscription

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query configuration information about a subscription.

API Function

This API is used by an NA to query the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
def querySingleSubscription(self, subscriptionId, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
subscriptionId	Mandatory	String	path	Identifies a subscription, which is obtained by calling or querying the subscription API.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

SubscriptionDTO structure

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

4.3.4.4 Querying Subscription in Batches

Typical Scenario

An NA can subscribe to different types of device change notifications on the IoT platform. The NA can call this API to query all subscription configurations of the current application or of a specified subscription type.

API Function

This API is used to query all subscription information of the current application or of a specified subscription type.

API Description

```
def queryBatchSubscriptions(self, qbsInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qbsInDTO	Mandatory	QueryBatchSubInDTO	query	For details, see QueryBatchSubInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryBatchSubInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String(256)	query	Identifies the application of the entity that subscribes to a device or rule.
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted:** device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the page size. The value is an integer greater than or equal to 1. The default value is **10**.

Response Parameters

QueryBatchSubOutDTO

Parameter	Type	Description
totalCount	long	Indicates the total number of records.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records on each page.
subscriptions	List<SubscriptionDTO>	Indicates the subscription information list. For details, see **SubscriptionDTO structure**.

SubscriptionDTO structure

Parameter	Type	Description
subscriptionId	String	Identifies a subscription.
notifyType	String	Indicates the notification type.
callbackUrl	String	Indicates the callback URL of the subscription.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100224	The resource exceeds 1000, please refinement query conditions.	The number of resources exceeds 1000. Recommended handling: Narrow down the filter criteria.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

4.3.4.5 Deleting a Subscription

Typical Scenario

If an NA does not need to receive a subscription notification message pushed by the IoT platform, the NA can call this API to delete the specified subscription configuration and cancel the subscription.

API Function

This API is used by an NA to delete the configuration information about a subscription by subscription ID on the IoT platform.

API Description

```
def deleteSingleSubscription(self, subscriptionId, appId, accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
subscriptionId	Mandatory	String(256)	path	Identifies a subscription.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **subscriptionId** is correct.

4.3.4.6 Deleting Subscriptions in Batches

Typical Scenario

If an NA does not need to receive subscription notification messages pushed by the IoT platform or a specified type of subscription notification messages, the NA can call this API to delete subscription configurations in batches and cancel the subscriptions.

API Function

This API is used to delete all subscriptions, subscriptions of a specified subscription type, or subscriptions of a specified callback URL in batches.

API Description

```
def deleteBatchSubscriptions(self, dbsInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dbsInDTO	Mandatory	DeleteBatchSubInDTO	body	For details, see the DeleteBatchSubInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeleteBatchSubInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String(256)	query	Identifies the application of the entity that subscribes to a device or rule.

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Optional	String(256)	query	Indicates the notification type based on which an NA can process messages. ● **bindDevice**: device binding ● **deviceAdded**: device addition ● **deviceInfoChanged**: device information change ● **deviceDataChanged**: device data change ● **deviceDatasChanged**: batch device data change ● **deviceDeleted:** device deletion ● **messageConfirm**: message confirmation ● **commandRsp**: command response ● **deviceEvent**: device event ● **serviceInfoChanged**: service information change ● **deviceModelAdded**: device model addition ● **deviceModelDeleted**: device model deletion ● **deviceDesiredPropertiesModifyStatusChanged**: device shadow modification status change ● **swUpgradeStateChangeNotify**: software upgrade status change notification ● **swUpgradeResultNotify**: software upgrade result notification ● **fwUpgradeStateChangeNotify**: firmware upgrade status change notification ● **fwUpgradeResultNotify**: firmware upgrade result notification
callbackUrl	Optional	String(256)	query	Indicates the callback URL of the subscription.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100225	The resource is not found	The resource does not exist. Recommended handling: Check whether **notifyType** is correct.

4.3.5 Message Push

NAs can subscribe to device information from the IoT platform. When the device information changes, the IoT platform pushes change notifications to the NAs. Then, the NAs distribute messages based on the notification type. This API must be used together with the **Subscription Management** API.

4.3.5.1 Pushing Device Registration Notifications

Typical Scenario

After an NA subscribes to device registration notifications (the notification type is **deviceAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when the NA registers a device on the IoT platform by calling the API for registering a directly connected device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device registration notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceAdded(self):
Class	PushMessageReceiver

Parameter Description

request.json

Parameter	Mandatory or Optional	Type	Location	Description
request.json	Mandatory	NotifyDeviceAddedDTO	body	For details, see NotifyDeviceAddedDTO structure.

NotifyDeviceAddedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceAdded**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
gatewayId	Optional	String	body	Uniquely identifies a gateway.
nodeType	Mandatory	String	body	Specifies the type of a device. ● **ENDPOINT** ● **GATEWAY** ● **UNKNOWN**

Parameter	Mandatory or Optional	Type	Location	Description
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.

Parameter	Mandatory or Optional	Type	Location	Description
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	Optional	String(256)	body	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.

Parameter	Mandatory or Optional	Type	Location	Description
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceAdded",
    "deviceId":"*****",
    "gatewayId":"*****",
    "nodeType":"GATEWAY",
```

```
    "deviceInfo":{
        "nodeId":"******",
        "name":null,
        "description":null,
        "manufacturerId":null,
        "manufacturerName":null,
        "mac":null,
        "location":null,
        "deviceType":null,
        "model":null,
        "swVersion":null,
        "fwVersion":null,
        "hwVersion":null,
        "protocolType":null,
        "bridgeId":null,
        "status":"OFFLINE",
        "statusDetail":"NOT_ACTIVE",
        "mute":null,
        "supportedSecurity":null,
        "isSecurity":null,
        "signalStrength":null,
        "sigVersion":null,
        "serialNumber":null,
        "batteryLevel":null
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.2 Pushing Device Binding Notifications

Typical Scenario

After an NA subscribes to device binding notifications (the notification type is **bindDevice**) on the IoT platform, the IoT platform sends a notification message to the NA when a directly connected device is connected to the IoT platform and bound to the NA.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device binding notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleBindDevice(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nbdDTO	Mandatory	NotifyBindDeviceDTO	body	For details, see NotifyBindDeviceDTO structure.

NotifyBindDeviceDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **bindDevice**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
resultCode	Mandatory	String	body	Indicates the binding result. The value options are **expired** and **succeeded**.
deviceInfo	Optional	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Identifies a device.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **Z-Wave**, **ZigBee**, and **WPS**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.

Parameter	Mandatory or Optional	Type	Location	Description
statusDetail	Optional	String(256)	body	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING

status	statusDetail
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"bindDevice",
    "deviceId":"*****",
    "resultCode":"succeeded",
    "deviceInfo":{
        "name":"Sensor_12",
        "manufacturer":"wulian",
        "deviceType":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
        "fwVersion": "...",
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smockdetector",
        "nodeType":"GATEWAY"
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.3 Pushing Device Information Change Notifications

Typical Scenario

After an NA subscribes to device information change notifications (the notification type is **deviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device configuration or status (such as manufacturer, location, version and online status) changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device information change notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceInfoChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
ndicDTO	Mandatory	NotifyDeviceInfoChangedDTO	body	For details, see NotifyDeviceInfo-ChangedDTO structure.

NotifyDeviceInfoChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceInfoChanged**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
deviceInfo	Mandatory	DeviceInfo	body	Indicates information about the device. For details, see **DeviceInfo structure**.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
nodeId	Mandatory	String(256)	body	Uniquely identifies the device. Generally, the MAC address, serial number, or IMEI is used as the node ID. **NOTE** When the IMEI is used as the node ID, the node ID varies depending on the chip provided by the manufacturer. ● The unique identifier of a Qualcomm chip is **urn:imei:xxxx**, where **xxxx** is the IMEI. ● The unique identifier of a HiSilicon chip is the IMEI. ● For details on the unique identifiers of chipsets provided by other manufacturers, contact the module manufacturers.
name	Optional	String(256)	body	Indicates the device name.
description	Optional	String(2048)	body	Indicates the device description.
manufacturerId	Optional	String(256)	body	Uniquely identifies a manufacturer.
manufacturerName	Optional	String(256)	body	Indicates the manufacturer name.
mac	Optional	String(256)	body	Indicates the MAC address of the device.
location	Optional	String(2048)	body	Indicates the device location.
deviceType	Optional	String(256)	body	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	Optional	String(256)	body	Indicates the device model.
swVersion	Optional	String(256)	body	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String(256)	body	Indicates the firmware version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
hwVersion	Optional	String(256)	body	Indicates the hardware version of the device.
protocolType	Optional	String(256)	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	Optional	String(256)	body	Identifies the bridge through which the device accesses the IoT platform.
status	Optional	String	body	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	Optional	String(256)	body	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.
mute	Optional	String	body	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	Optional	String	body	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	Optional	String	body	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	Optional	String(256)	body	Indicates the signal strength of the device.
sigVersion	Optional	String(256)	body	Indicates the SIG version of the device.

Parameter	Mandatory or Optional	Type	Location	Description
serialNumber	Optional	String(256)	body	Indicates the serial number of the device.
batteryLevel	Optional	String(256)	body	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType ":"deviceInfoChanged",
    "deviceId":"******************************",
    "gatewayId":"*************",
    "deviceInfo":{
        "name":"Sensor_12",
        "manufacturer":"wulian",
        "type":90,
        "model":"90",
        "mac":"****************",
        "swVersion": "...",
        "fwVersion": "...",
        "hwVersion": "...",
        "protocolType":"zigbee",
        "description":"smock detector"
    }
}
```


Response Example

```
response:
Status Code: 200 OK
```

4.3.5.4 Pushing Device Data Change Notifications

Typical Scenario

After an NA subscribes to device data change notifications (the notification type is **deviceDataChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of a single service attribute.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device data change notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceDataChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nddcDTO	Mandatory	NotifyDeviceDataChangedDTO	body	For details, see NotifyDeviceDataChangedDTO structure.

NotifyDeviceDataChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDataChanged**.
requestId	Optional	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Uniquely identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
service	Mandatory	DeviceService	body	Indicates service data of the device. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.
eventTime	Mandatory	String	body	Indicates the event occurrence time in the format of yyyymmddThhmissZ, for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
```

```
Body:
{
    "notifyType":"deviceDataChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":{
        "serviceId":"Brightness",
        "serviceType":"Brightness",
        "data":{
            "brightness":80
        },
        "eventTime":"20170311T163657Z"
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.5 Pushing Batch Device Data Change Notifications

Typical Scenario

After an NA subscribes to batch device data change notifications (the notification type is **deviceDatasChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device reports data of multiple service attributes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to batch device data change notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceDatasChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nddscDTO	Mandatory	NotifyDeviceDatasChangedDTO	body	For details, see NotifyDeviceDatasChangedDTO structure.

NotifyDeviceDatasChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDatasChanged**.
requestId	Mandatory	String	body	Indicates the sequence number of the message, which uniquely identifies the message.
deviceId	Mandatory	String	body	Uniquely identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
services	Mandatory	List<DeviceService>	body	Indicates a service list. For details, see **DeviceService structure**.

DeviceService structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
data	Mandatory	ObjectNode	body	Indicates service data information.

Parameter	Mandatory or Optional	Type	Location	Description
eventTime	Mandatory	String	body	Indicates the time when an event is reported in the format of yyyyMMddTHHmmssZ, for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDatasChanged",
    "requestId":"*************",
    "deviceId":"********************************",
    "gatewayId":"*************",
    "service":[
        {
            "serviceId":"Brightness",
            "serviceType":"Brightness",
            "data":{
                "brightness":80
            },
            "eventTime":"20170311T163657Z"
        },
        {
            "serviceId":"Color",
            "serviceType":"Color",
            "data":{
                "value":"red"
            },
            "eventTime":"20170311T163657Z"
        }
    ]
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.6 Pushing Device Service Information Change Notifications

Typical Scenario

After an NA subscribes to device service information change notifications (the notification type is **serviceInfoChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device to modify the device service information.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device service information change notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleServiceInfoChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nsicDTO	Mandatory	NotifyServiceInfoChangedDTO	body	For details, see NotifyServiceInfo-ChangedDTO structure.

NotifyServiceInfoChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	Enum	body	Indicates the notification type. The value is **serviceInfoChanged**.
deviceId	Mandatory	String	body	Uniquely identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.
serviceId	Mandatory	String	body	Identifies a service.
serviceType	Mandatory	String	body	Indicates the service type.
serviceInfo	Mandatory	ServiceInfo	body	Indicates the device service information, which is incrementally reported. For details, see **ServiceInfo structure**.

ServiceInfo structure

Parameter	Mandatory or Optional	Type	Location	Description
muteCmds	Optional	List<String>	body	Indicates the device command list.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"serviceInfoChanged",
    "deviceId":"*******",
    "serviceId":"*******",
    "serviceType":"*******",
    "gatewayId":"*******",
    "serviceInfo":{
        "muteCmds":"VIDEO_RECORD"
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```


4.3.5.7 Pushing Device Deletion Notifications

Typical Scenario

After an NA subscribes to device deletion notifications (the notification type is **deviceDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when the device is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device deletion notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceDeleted(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nddDTO	Mandatory	NotifyDeviceDeletedDTO	body	For details, see NotifyDeviceDeletedDTO structure.

NotifyDeviceDeletedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDeleted**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
gatewayId	Mandatory	String	body	Uniquely identifies a gateway.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDeleted",
    "deviceId":"******************************",
    "gatewayId":"**********"
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.8 Pushing Device Acknowledgment Notifications

Typical Scenario

After an NA subscribes to device acknowledgment notifications (the notification type is **messageConfirm**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command acknowledgment message (for example, the command is delivered or executed).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device acknowledgment notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.

3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleMessageConfirm(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nmcDTO	Mandatory	NotifyMessageConfirmDTO	body	For details, see NotifyMessageConfirmDTO structure.

NotifyMessageConfirmDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **messageConfirm**.
header	Mandatory	MessageConfirmHeader	body	For details, see **MessageConfirmHeader structure**.
body	Mandatory	ObjectNode	body	Based on the service definition, an acknowledgment message can carry information such as status change.

MessageConfirmHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
status	Mandatory	String(1-32)	body	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been received. ● **executed**: The command has been executed.
timestamp	Mandatory	String(1-32)	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"messageConfirm",
    "header":{
        "requestId":"************",
        "from":"************",
        "to":"************",
        "status":"delivered",
        "timestamp":"20151212T121212Z"
    },
    "body":{

    }
}
```


Response Example

```
response:
Status Code: 200 OK
```

4.3.5.9 Pushing Device Command Response Notifications

Typical Scenario

After an NA subscribes to device command response notifications (the notification type is **commandRsp**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform delivers a command to the device and the device returns a command response message (for example, the command execution succeeds or fails).

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device command response notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleCommandRsp(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
ncrDTO	Mandatory	NotifyCommandRspDTO	body	For details, see NotifyCommandRspDTO structure.

NotifyCommandRspDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **commandRsp**.
header	Mandatory	CommandRspHeader	body	For details, see **CommandRspHeader structure**.
body	Mandatory	ObjectNode	body	Indicates the content of the message body of the response command.

CommandRspHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Mandatory	String(1-128)	body	Indicates the sequence number of the message, which uniquely identifies the message.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**
to	Mandatory	String(1-128)	body	Indicates the address of the message recipient. The value is that of **from** in the request, for example, the user ID of the NA.
deviceId	Mandatory	String	body	Uniquely identifies a device.
serviceType	Mandatory	String	body	Indicates the type of the service to which a command belongs.
method	Mandatory	String(1-128)	body	Indicates a stored response command, for example, **INVITE-RSP**.

Response Parameters

```
Status Code: 200 OK
```


Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"commandRsp",
    "header":{
        "requestId":"***********",
        "from":"***********",
        "to":"***********",
        "deviceId":"***********",
        "serviceType":"Camera",
        "method":"MUTE_COMMANDS"
    },
    "body":{

    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.10 Pushing Device Event Notifications

Typical Scenario

After an NA subscribes to device event notifications (the notification type is **deviceEvent**) on the IoT platform, the IoT platform sends a notification message to the NA when the IoT platform receives the event message reported by the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device event notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceEvent(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
ndeDTO	Mandatory	NotifyDeviceEventDTO	body	For details, see NotifyDeviceEventDTO structure.

NotifyDeviceEventDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceEvent**.
header	Mandatory	DeviceEventHeader	body	For details, see **DeviceEventHeader structure**.
body	Mandatory	Object Node	body	Indicates the content of the message body of the response command.

DeviceEventHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
eventType	Mandatory	String(1-32)	body	Indicates the event type.
from	Mandatory	String(1-128)	body	Indicates the address of the message sender. ● Request initiated by a device: **/devices/{deviceId}** ● Request initiated by a device service: **/devices/{deviceId}/services/{serviceId}**

Parameter	Mandatory or Optional	Type	Location	Description
timestamp	Mandatory	String (1-32)	body	Indicates the timestamp. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
eventTime	Mandatory	String (1-32)	body	Indicates the time when an event is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
serviceType	Mandatory	String	body	Indicates the type of the service to which a command belongs.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceEvent",
    "header":{
        "eventType":"*******",
        "from":"/devices/{deviceId}/services/{serviceId}",
        "deviceId":"*******",
        "serviceType":"*******",
        "timestamp":"20151212T121212Z",
        "eventTime":"20151212T121212Z"
    },
    "body":{
        "usedPercent":80
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.11 Pushing Device Model Addition Notifications

Typical Scenario

After an NA subscribes to device model addition notifications (the notification type is **deviceModelAdded**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is added on the IoT platform

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model addition notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceModelAdded(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
ndmaDTO	Mandatory	NotifyDeviceModelAddedDTO	body	For details, see NotifyDeviceModelAddedDTO structure.

NotifyDeviceModelAddedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceModelAdded**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the operator name.
manufacturerId	Mandatory	String	body	Identifies the operator.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceType":"*************",
    "manufacturerName":"wulian",
    " manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"zigbee"
}
```

Response Example

```
response:
Status Code: 200 OK
```


4.3.5.12 Pushing Device Model Deletion Notifications

Typical Scenario

After an NA subscribes to device model deletion notifications (the notification type is **deviceModelDeleted**) on the IoT platform, the IoT platform sends a notification message to the NA when a device profile file is deleted from the IoT platform.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device model deletion notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceModelDeleted(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
ndmdDTO	Mandatory	NotifyDeviceModelDeletedDTO	body	For details, see NotifyDeviceModelDeletedDTO structure.

NotifyDeviceModelDeletedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceModelDeleted**.
appId	Mandatory	String	body	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceType	Mandatory	String	body	Indicates the device type.
manufacturerName	Mandatory	String	body	Indicates the operator name.
manufacturerId	Mandatory	String	body	Identifies the operator.
model	Mandatory	String	body	Indicates the device model.
protocolType	Mandatory	String	body	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceModelAdded",
    "appId":"*****",
    "deviceTyp ":"*************",
    " manufacturerName":"*************",
    "manufacturerId ":"*************",
    "model":"*************",
    "protocolType":"*************"
}
```

Response Example

```
response:
Status Code: 200 OK
```


4.3.5.13 Pushing Device Shadow Status Change Notifications

Typical Scenario

After an NA subscribes to device shadow status change notifications (the notification type is **deviceDesiredPropertiesModifyStatusChanged**) on the IoT platform, the IoT platform sends a notification message to the NA when the device shadow on the IoT platform succeeds or fails to synchronize data to the device.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to device shadow status change notifications.

Note

1. When **Subscribing to Service Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleDeviceDesiredStatusChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nddscDTO	Mandatory	NotifyDeviceDesiredStatusChangedDTO	body	For details, see NotifyDeviceDesiredStatusChangedDTO structure.

NotifyDeviceDesiredStatusChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **deviceDesiredPropertiesModifyStatusChanged**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
serviceId	Mandatory	String	body	Identifies a service.
properties	Mandatory	ObjectNode	body	Indicates data attributes of a device shadow.
status	Mandatory	String	body	Indicates the status. The value options are **DELIVERED** and **FAILED**.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"deviceDesiredPropertiesModifyStatusChanged",
    "deviceId":"******",
    "serviceId":"Device",
    "properties":{
        "Model Number":1,
        "Serial Number":2,
        "Firmware Version":"v1.1.0"
    },
    "status":"DELIVERED"
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.14 Pushing Software Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to software upgrade status change notifications (the notification type is **swUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the software upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade status change notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleSwUpgradeStateChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nsuscDTO	Mandatory	NotifySwUpgradeStateChangedDTO	body	For details, see NotifySwUpgradeStateChangedDTO structure.

NotifySwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeStateChangeNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade sub-task.
swUpgradeState	Mandatory	String	body	Indicates the software upgrade status. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"swUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"***********",
    "subOperationId":"***********",
    "swUpgradeState":"downloading"
}
```

Response Example

```
response:
Status Code: 200 OK
```


4.3.5.15 Pushing Software Upgrade Result Notifications

Typical Scenario

After an NA subscribes to software upgrade result change notifications (the notification type is **swUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a software upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to software upgrade result notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleSwUpgradeResult(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nsurDTO	Mandatory	NotifySwUpgradeResultDTO	body	For details, see NotifySwUpgradeResultDTO structure.

NotifySwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **swUpgradeResultNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a software upgrade task.
subOperationId	Mandatory	String	body	Identifies a software upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current software version of the device.
targetVersion	Mandatory	String	body	Indicates the target software version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source software version of the device.
swUpgradeResult	Mandatory	String	body	Indicates the software upgrade result. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the upgrade time.
resultDesc	Mandatory	String	body	Indicates the upgrade result description.
errorCode	Mandatory	String	body	Indicates a status error code reported by the device.
description	Mandatory	String	body	Indicates the description of the cause of error.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
```



```
Header:
Content-Type:application/json
Body:
{
    "notifyType":"swUpgradeResultNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"***********",
    "subOperationId":"***********",
    "curVersion":"1.3",
    "targetVersion":"1.5",
    "sourceVersion":"1.0",
    "swUpgradeResult":"SUCCESS",
    "upgradeTime":"***",
    "resultDesc":"***",
    "errorCode":"***",
    "description":"***"
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.16 Pushing Firmware Upgrade Status Change Notifications

Typical Scenario

After an NA subscribes to firmware upgrade status change notifications (the notification type is **fwUpgradeStateChangeNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when the firmware upgrade status changes.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade status change notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleFwUpgradeStateChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nfuscDTO	Mandatory	NotifyFwUpgradeStateChangedDTO	body	For details, see NotifyFwUpgradeStateChangedDTO structure.

NotifyFwUpgradeStateChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeStateChangeNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade subtask.
step	Mandatory	String	body	Indicates the firmware upgrade status. The value options are **0**, **1**, **2**, and **3**.
stepDesc	Mandatory	String	body	Indicates the upgrade status description. ● **downloading**: The device is downloading the software package. ● **downloaded**: The device has finished downloading the software package. ● **updating**: The device is being upgraded. ● **idle**: The device is in the idle state.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeStateChangeNotify",
    "deviceId":"******",
    "appId":"***********",
    "operationId":"**************",
    "subOperationId":"***********",
    "step":"1",
    "stepDesc":"downloading"
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.17 Pushing Firmware Upgrade Result Notifications

Typical Scenario

After an NA subscribes to firmware upgrade result change notifications (the notification type is **fwUpgradeResultNotify**) on the IoT platform, the IoT platform sends a notification message to the NA when a firmware upgrade task is complete.

API Function

This API is used by the IoT platform to push notification messages to an NA that has subscribed to firmware upgrade result notifications.

Note

1. When **Subscribing to Management Data of the IoT Platform**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA, the NA must implement the original callback API. For details on the API content, see **Message Push** in the *Huawei IoT Platform Northbound API Reference*.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver
Callback API	**def** handleFwUpgradeResult(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nfurDTO	Mandatory	NotifyFwUpgradeResultDTO	body	For details, see NotifyFwUpgradeResultDTO structure.

NotifyFwUpgradeResultDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
notifyType	Mandatory	String	body	Indicates the notification type. The value is **fwUpgradeResultNotify**.
deviceId	Mandatory	String	body	Uniquely identifies a device.
appId	Mandatory	String	body	Identifies the application to which the device belongs.
operationId	Mandatory	String	body	Identifies a firmware upgrade task.
subOperationId	Mandatory	String	body	Identifies a firmware upgrade sub-task.
curVersion	Mandatory	String	body	Indicates the current firmware version of the device.
targetVersion	Mandatory	String	body	Indicates the target firmware version to which the device is to be upgraded.
sourceVersion	Mandatory	String	body	Indicates the source firmware version of the device.
Status	Mandatory	String	body	Indicates the upgrade result. ● **SUCCESS** ● **FAIL**

Parameter	Mandatory or Optional	Type	Location	Description
statusDesc	Mandatory	String	body	Indicates the upgrade result description. ● **SUCCESS**: The device upgrade is successful. ● **FAIL**: The device upgrade fails.
upgradeTime	Mandatory	String	body	Indicates the firmware upgrade duration.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "notifyType":"fwUpgradeResultNotify",
    "deviceId":"******",
    "appId":"**********",
    "operationId":"**********",
    "subOperationId":"**********",
    "curVersion":"1.6",
    "targetVersion":"1.6",
    "sourceVersion":"1.3",
    "status":"SUCCESS",
    "statusDesc":"****",
    "upgradeTime":"****"
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.5.18 NB-IoT Device Command Status Change Notification

Typical Scenario

When an NA creates a device command with the callback URL specified, the IoT platform pushes a notification message to the NA if the command status changes (failed, successful, timeout, sent, or delivered).

API Function

The IoT platform pushes notification messages to NAs when the command status changes.

Note

1. When **Creating Device Commands**, an NA must subscribe to the specified callback address in the API description. The server and port in the callback address are the public IP address and specified port of the NA server.
2. An NA receives the content of a push message by inheriting the PushMessageReceiver class and rewriting the callback API.
3. If the callback address is not the address of the NA server, the NA must implement the original callback API.

API Description

Callback URL	https://server:port/v1.0.0/messageReceiver/cmd
Callback API	**def** handleNBCommandStateChanged(self)
Class	PushMessageReceiver

Parameter Description

The input parameter is request.json.

Parameter	Mandatory or Optional	Type	Location	Description
nNBcscDTO	Mandatory	NotifyNBCommandStatusChangedDTO	body	For details, see NotifyNBCommandStatusChangedDTO structure.

NotifyNBCommandStatusChangedDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	body	Uniquely identifies a device.
commandId	Mandatory	String	body	Identifies a command, which is generated by the IoT platform during device creation.
result	Mandatory	NBCommandResult	body	For details, see **NBCommandResult structure**.

NBCommandResult structure

Parameter	Mandatory or Optional	Type	Location	Description
resultCode	Mandatory	String	body	Indicates the command status result. ● **SENT**: The IoT platform has delivered a command to the device but has not received a response from the device. ● **DELIVERED**: The IoT platform receives a response from a device. ● **SUCCESS**: The IoT platform receives a command result and the result is success. ● **FAIL**: The IoT platform receives a command result and the result is a failure.
resultDetail	Mandatory	ObjectNode	body	Indicates the user-defined fields carried in the command result.

Response Parameters

```
Status Code: 200 OK
```

Request Example

```
Method: POST
request: {callbackUrl}
Header:
Content-Type:application/json
Body:
{
    "deviceId":"*********",
    "commandId":"***********",
    "result":{
        "resultCode":"DELIVERED",
        "resultDetail":"None"
    }
}
```

Response Example

```
response:
Status Code: 200 OK
```

4.3.6 Command Delivery (NB-IoT Commands)

4.3.6.1 Creating Device Commands

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform provides two command delivery modes:

- Immediate delivery: The IoT platform delivers commands to devices immediately after receiving the commands. This ensures real-time performance but does not ensure serialization.
- Pending delivery: After receiving commands, the IoT platform caches the commands. When the devices are reachable, the IoT platform delivers the commands in sequence. Specifically, the IoT platform delivers the latter command only after receiving the response of the previous command (which is the ACK automatically replied by the module) to ensure serialization instead of real-time performance.

API Function

This API is used by NAs to deliver commands to devices. Immediate delivery and pending delivery are supported on the IoT platform.

API Description

```
def postDeviceCommand(self, pdcInDTO, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
pdcInDTO	Mandatory	PostDeviceCommandInDTO	body	For details, see PostDeviceCommandInDTO structure.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

PostDeviceCommandInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(64)	body	Uniquely identifies the device that delivers the command.
command	Mandatory	CommandDTOV4	body	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	Optional	String(1024)	body	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Optional	Integer(>=0)	body	Indicates the command expiration time, in seconds. That is, the validity period of the created device command is **expireTime** seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (86400 seconds x 2). If this parameter is set to **0**, the IoT platform will deliver the command to the specific device immediately regardless of the command mode set on the IoT platform (if the device is sleeping or the link has aged, the device cannot receive the command, the IoT platform cannot receive any response from the device, and the command times out in the end).
maxRetransmit	Optional	Integer(0~3)	body	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command. The value of this parameter must be the same as **serviceId** defined in the profile.
method	Mandatory	String(1-128)	body	Indicates the command name. The value of this parameter must be the same as the command name defined in the profile file.
paras	Mandatory	ObjectNode	body	Indicates a command parameter in the jsonString format. The value consists of key-value pairs. Each key is the **paraName** parameter in commands in the profile file. The specific format depends on the application and device. If no parameter is defined in the command in the profile file, left it blank, that is, "paras": {}.

Response Parameters

PostDeviceCommandOutDTO structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).

Parameter	Type	Description
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0~3)	Indicates the maximum number of times the command can be retransmitted.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	100223	Command counts has reached the upLimit.	The number of cached commands reaches the limit. The number of commands in the **PENDING** state does not exceed the limit. The default value is **20**. Recommended handling: If the commands cached on the IoT platform need to be executed, enable the device to report data to trigger delivery of the cache commands. If a command cached on the IoT platform does not need to be executed, call the API used for modifying device commands V4 to change the state of the command from **PENDING** to **CANCELED**.

HTTP Status Code	Error Code	Error Description	Remarks
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	100612	Device is zombie.	The device is a zombie device. (The interval between the current system time and the time when the device went online exceeds the threshold. The default value is seven days.) Recommended handling: Run the command again after the device goes online.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.6.2 Querying Device Commands

Typical Scenario

After an NA delivers a command to a device, the NA can call this API to query the status and content of the delivered command on the IoT platform to check the command execution status.

API Function

This API is used by an NA to query the status and content of delivered commands on the IoT platform. All the commands delivered by the current application in a specified period or all the commands delivered to a specified device can be queried.

API Description

```
def queryDeviceCommand(self, qdcInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcInDTO	Mandatory	QueryDeviceCommandInDTO	query	For details, see QueryDeviceCommandInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCommandInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
pageSize	Optional	Integer(>=1&&<=1000)	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
deviceId	Optional	String(64)	query	Identifies the device whose commands are to be queried.
startTime	Optional	String	query	Indicates the start time. Commands delivered later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Commands delivered earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
appId	Optional	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCommandOutDTO structure

Parameter	Type	Description
pagination	Pagination	Indicates pagination information. For details, see **Pagination structure**.
data	List<DeviceCommandRespV4>	Indicates the device command list. For details, see **DeviceCommandRespV4 structure**.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalSize	long	Indicates the total number of records.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.

Parameter	Type	Description
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0~3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.

HTTP Status Code	Error Code	Error Description	Remarks
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: ● Ensure that neither **startTime** nor **endTime** is null and the value of **endTime** is later than that of **startTime**. ● Ensure that **pageNo** is not null and the value of **pageNo** is greater than **0**. ● Ensure that **pageSize** is not null and the value of **pageSize** is greater than **1**.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.6.3 Modifying Device Commands

Typical Scenario

NAs can call this API to modify the status of commands that have not been canceled, expired, or executed. Currently, the status of such commands can only be changed to **Canceled**.

API Function

This API is used by NAs to modify the status of commands. Currently, the status of commands can only be changed to **Canceled**. That is, the commands are revoked.

API Description

```
def updateDeviceCommand(self, udcInDTO, deviceCommandId, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
udcInDTO	Mandatory	UpdateDeviceCommandInDTO	body	For details, see UpdateDeviceCommandInDTO structure.
deviceCommandId	Mandatory	String	path	Identifies the command whose status is to be modified. The value of this parameter is obtained after the API used for creating device commands is called.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

UpdateDeviceCommandInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
status	Mandatory	String	body	Indicates the command execution result. The value can be **CANCELED**, which indicates that the command is revoked.

Response Parameters

UpdateDeviceCommandOutDTO

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.

Parameter	Type	Description
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0~3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Location	Description
serviceId	Mandatory	String(1-64)	body	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	body	Indicates the command name.
paras	Optional	Object	body	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the IoT platform is normal.

HTTP Status Code	Error Code	Error Description	Remarks
200	100431	The serviceType is not exist.	The service type does not exist. Recommended handling: ● Check whether the profile file of the device has been uploaded to the IoT platform. ● Check whether the request parameters are correct and whether **serviceId** exists in the profile file.
200	100434	The device command is not existed.	The device command does not exist. Recommended handling: Check whether the device command ID in the request is correct.
200	100435	The device command already canceled, expired or executed, Cannot cancel.	The device command has been canceled, expired, or executed. It cannot be canceled.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in dataBase is abnomal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

HTTP Status Code	Error Code	Error Description	Remarks
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
500	101016	Get iotws address failed.	Failed to obtain the IoTWS address. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	101017	Get newCallbackUrl from oss failed.	Obtaining a new callback URL from the OSS fails. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.6.4 Creating Device Command Revocation Tasks

Typical Scenario

After an NA delivers commands to a device, the IoT platform does not deliver the commands to the device for execution (the commands are in the **DEFAULT** state) if the commands are in queue or the device is offline. In this case, the NA can call this API to revoke all the undelivered commands of a specified device. Commands that have been delivered cannot be revoked.

API Function

This API is used by an NA to create a command revocation task to revoke all undelivered commands (that is, commands in the **DEFAULT** state) with the specified device ID on the IoT platform.

API Description

```
def createDeviceCmdCancelTask(self, cdcctInDTO, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cdcctInDTO	Mandatory	CreateDeviceCmdCancelTaskInDTO	body	For details, see CreateDeviceCmdCancelTaskInDTO structure.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateDeviceCmdCancelTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String(1-64)	body	Identifies the device whose commands are to be revoked. The revocation task will revoke all commands delivered to this device.

Response Parameters

CreateDeviceCmdCancelTaskOutDTO structure

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Identifies the device whose commands are to be revoked by the revocation task.

Parameter	Type	Description
status	String	Indicates the status of the command revocation task. ● **WAITTING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<DeviceCommandRespV4>	Indicates the revoked device command list. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).

Parameter	Type	Description
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0~3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.6.5 Querying Command Revocation Tasks

Typical Scenario

After delivering a command revocation command to a device, an NA can call this API to query the execution status of the command revocation task.

API Function

This API is used by an NA to query the information and status of one or more command revocation tasks based on specified conditions on the IoT platform.

API Description

```
def queryDeviceCmdCancelTask(self, qdcctInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcctInDTO	Mandatory	QueryDeviceCmdCancelTaskInDTO2	query	For details, see QueryDeviceCmdCancelTaskInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCmdCancelTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer(>=0)	query	Indicates the page number. The value is greater than or equal to 0. The default value is **0**.
pageSize	Optional	Integer(>=1 &&<=1000)	query	Indicates the number of records to be displayed on each page. The value range is 1 - 1000. The default value is **1000**.
taskId	Optional	String	query	Identifies a command revocation task.
deviceId	Optional	String	query	Identifies the device whose commands are to be revoked by the revocation task.
status	Optional	String	query	Indicates the status of the command revocation task.
startTime	Optional	String	query	Indicates the start time. Revocation tasks created later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Revocation tasks created earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String	query	If the command belongs to the current application, set this parameter to **null**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

QueryDeviceCmdCancelTaskOutDTO structure

Parameter	Type	Description
pagination	Pagination	Indicates pagination information. For details, see **Pagination structure**.
data	List<DeviceCommandCancelTaskRespV4>	Indicates the device command list. For details, see **DeviceCommandCancelTaskRespV4 structure**.

Pagination structure

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalSize	long	Indicates the total number of records, that is, the total number of commands queried in the command revocation task.

DeviceCommandCancelTaskRespV4 structure

Parameter	Type	Description
taskId	String(1-64)	Identifies a command revocation task.
appId	String(1-64)	Identifies the application to which the command revocation task belongs.
deviceId	String(1-64)	Identifies the device whose commands are to be revoked by the revocation task.

Parameter	Type	Description
status	String	Indicates the status of the command revocation task. ● **WAITTING**: The task is waiting to be executed. ● **RUNNING**: The task is being executed. ● **SUCCESS**: The task has been successfully executed. ● **FAILED**: The task fails to be executed. ● **PART_SUCCESS**: Task execution partially succeeds.
totalCount	Integer	Indicates the total number of revoked commands.
deviceCommands	List<DeviceCommandRespV4>	Indicates a list of device commands to be revoked by the revocation task. For details, see **DeviceCommandRespV4 structure**.

DeviceCommandRespV4 structure

Parameter	Type	Description
commandId	String(1-64)	Identifies a device command.
appId	String(1-64)	Uniquely identifies an NA. This parameter is used to identify an NA that can call open APIs provided by the IoT platform.
deviceId	String(1-64)	Uniquely identifies the device that delivers the command.
command	CommandDTOV4	Indicates information about the delivered command. For details, see **CommandDTOV4 structure**.
callbackUrl	String(1024)	Indicates the URL for receiving command status change notifications. When the command status changes, such as execution failure, execution success, timeout, sending, or sent, the NA is notified.
expireTime	Integer(>=0)	Indicates the command expiration time, in units of seconds. The command will not be delivered after the specified time elapses. The default validity period is 48 hours (86400 seconds x 2).

Parameter	Type	Description
status	String	Indicates the status of the command. ● **DEFAULT**: The command has not been delivered. ● **EXPIRED**: The command has expired. ● **SUCCESSFUL**: The command has been successfully executed. ● **FAILED**: The command fails to be executed. ● **TIMEOUT**: Command execution times out. ● **CANCELED**: The command has been canceled.
result	ObjectNode	Indicates the detailed command execution result.
creationTime	String(20)	Indicates the time when the command is created.
executeTime	String(20)	Indicates the time when the command is executed.
platformIssuedTime	String(20)	Indicates the time when the IoT platform sends the command.
deliveredTime	String(20)	Indicates the time when the command is delivered.
issuedTimes	Integer(>=0)	Indicates the number of times the IoT platform delivers the command.
maxRetransmit	Integer(0~3)	Indicates the maximum number of times the command can be retransmitted.

CommandDTOV4 structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String(1-64)	Identifies the service corresponding to the command.
method	Mandatory	String(1-128)	Indicates the command name.
paras	Optional	Object	Indicates a JSON string of command parameters. The specific format is negotiated by the NA and the device.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100220	Get AppKey from header failed.	Failed to obtain the appKey. Recommended handling: Check whether **appId** is carried in the API request header.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.7 Command Delivery (Non-NB-IoT Commands)

4.3.7.1 Calling Device Services

Typical Scenario

The device profile file defines commands that the IoT platform can deliver to a device. When an NA needs to configure or modify the service attributes of a device, the NA can call this API to deliver commands to the device.

The IoT platform does not cache commands but delivers commands directly. When a device is offline, the commands fail to be delivered. The formats of the delivered command need to be defined by the NAs and devices. The IoT platform encapsulates and transparently transmits the commands over this API.

NOTE

Currently, this API can be used to deliver commands only to gateways equipped with the IoT Agent or AgentLite to control non-directly connected devices under the gateways.

API Function

This API is used to immediately deliver commands to gateways equipped with the IoT Agent or AgentLite to control the gateways. This API applies to devices registered with the current application.

API Description

```
def invokeDeviceService(self, commandDTO, deviceId, serviceId, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
commandDTO	Mandatory	CommandDTO	body	For details, see **CommandDTO structure**.
deviceId	Mandatory	String(1-64)	path	Uniquely identifies a device.
serviceId	Mandatory	String(1-64)	path	Uniquely identifies a service.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CommandDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
header	Mandatory	CommandNA2CloudHeader	body	For details, see **CommandNA2CloudHeader structure**.
body	Optional	Object	body	Indicates the message body. The content of the JsonObject is a list of key-value pairs. Every key is the **paraName** of a command defined in the profile.

CommandNA2CloudHeader structure

Parameter	Mandatory or Optional	Type	Location	Description
requestId	Optional	String(0-128)	body	Identifies a command. The value of this parameter must be unique.
mode	Mandatory	Enum	body	Indicates whether an ACK message is required. ● **NOACK**: No ACK message is required. ● **ACK**: An ACK message is required. ● Other values: invalid
from	Optional	String(128)	body	Indicates the address of the message sender. ● Request initiated by an application: **/users/{userId}** ● Request initiated by an NA: **/{serviceName}** ● Request initiated by the IoT platform: **/cloud/{serviceName}**
toType	Optional	Enum	body	Indicates the type of the message recipient. The value options are **CLOUD** and **GATEWAY**.
to	Optional	String(128)	body	Indicates the address of the message recipient.
method	Mandatory	String(1-32)	body	Indicates the command name. For example, a **DISCOVERY** command is used to discover non-directly connected devices, and a **REMOVE** command is used to delete non-directly connected devices.
callbackURL	Optional	String(1024)	body	Indicates the callback URL.

Response Parameters

InvokeDeviceServiceOutDTO

Parameter	Type	Description
status	String(128)	Indicates the command status. ● **sent**: The command has been sent. ● **delivered**: The command has been delivered. This value is returned when **toType** is set to **CLOUD**. ● **failed**: The command fails to be delivered. This value is returned when **toType** is set to **CLOUD**.
timestamp	String(128)	Indicates the timestamp used for sending a command. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
requestId	String(128)	Identifies a device command. ● When **toType** is set to **GATEWAY**, if **requestId** is carried in a request, the response carries the same **requestId** as the request; if **requestId** is not carried in a request, the IoT platform allocates a sequence number for the response. ● When **toType** is set to **CLOUD**, the value of this parameter is null.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
200	100428	The device is not online.	The device is not online. Recommended handling: Check whether the connection between the device and the gateway is normal.
200	100432	The device command is muted.	The device command is muted. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
400	102203	CommandName is invalid.	The command name is invalid. Recommended handling: Check whether the command carried in the API request parameter **method** is correct.
403	100450	The gateway is not online.	The gateway is offline. Recommended handling: Check whether the connection between the gateway and the IoT platform is normal.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100444	The serviceType is not exist.	The service type does not exist. Recommended handling: Check whether the service type carried in the API request parameter **toType** is correct.
500	100001	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	100023	The data in database is abnormal.	The database is abnormal. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.
503	100501	Congestion occurs, and the current network has been flow-controlled	Congestion occurs. The current network is under flow control.

4.3.8 Data Collection

The IoT platform allows NAs to query basic information about a device and view historical data reported by the device by hour, day, or month.

4.3.8.1 Querying Information About a Device

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of a device that has been registered with the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information of a specified device based on the device ID on the IoT platform, such as configuration, status and service attributes.

API Description

```
def querySingleDeviceInfo(self, deviceId, select, appId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
deviceId	Mandatory	String	path	Identifies a device. The device ID is allocated by the IoT platform during device registration.
select	Optional	String	query	Indicates the query condition. The value can be **IMSI**.
appId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QuerySingleDeviceInfoOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device.
gatewayId	String(256)	Identifies a gateway.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.
deviceInfo	DeviceInfo	Indicates information about the device. For details, see **DeviceInfo structure**.
services	List<DeviceService>	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	String(256)	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.

Parameter	Type	Description
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.

Parameter	Type	Description
serviceInfo	ServiceInfo	Indicates service information of the device. For details, see **ServiceInfo structure**.
data	ObjectNode(2097152)	Indicates an attribute-value pair (AVP).
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the device command list.

Error Codes

HTTP Status Code	Error Codes	Error Description	Remarks
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Codes	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	100403	The device is not existed.	The device does not exist. Recommended handling: Check whether **deviceId** is correct.
404	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.8.2 Querying a Device Information List

Typical Scenario

If an NA needs to view detailed information (such as the manufacturer, model, version, status, and service attributes) of multiple devices that have been registered with the IoT platform, the NA can call this API to obtain the information.

API Function

This API is used by an NA to query detailed information (such as configuration, status and service attributes) of multiple devices based on specified conditions on the IoT platform.

API Description

```
def queryBatchDevicesInfo(self, qbdiInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qbdiInDTO	Mandatory	QueryBatchDevicesInfoInDTO	query	For details, see QueryBatchDevicesInfoInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryBatchDevicesInfoInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
gatewayId	Optional	String	query	Identifies a gateway.
nodeType	Optional	String	query	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
deviceType	Optional	String	query	Specifies the type of a device.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.
status	Optional	String	query	Indicates the device status. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline. ● **ABNORMAL**: The device is abnormal.
startTime	Optional	String	query	Indicates the start time. Records with the device registration time later than the specified start time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Records with the device registration time earlier than the specified end time are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
sort	Optional	String	query	Indicates the sorting mode of queried records. ● **ASC**: Records are sorted in ascending order of device registration time. ● **DESC**: Records are sorted in descending order of device registration time. The default value is **DESC**.
select	Optional	String	query	Indicates the record to be returned. The value can be **IMSI**.

Response Parameters

QueryBatchDevicesInfoOutDTO

Parameter	Type	Description
totalCount	long	Indicates the number of queried records.

Parameter	Type	Description
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records on each page.
devices	List< QuerySingleDeviceInfoOutDTO>	Indicates the device pagination information list. For details, see **QuerySingleDeviceInfoOutDTO structure**.

QuerySingleDeviceInfoOutDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device.
gatewayId	String(256)	Identifies a gateway.
nodeType	Enum	Indicates the node type. The value options are **ENDPOINT**, **GATEWAY**, and **UNKNOW**.
createTime	String(256)	Indicates the time when the device is created. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
lastModifiedTime	String(256)	Indicates the last time when the device is modified.
deviceInfo	DeviceInfoQueryDTO	Indicates information about the device. For details, see **DeviceInfo structure**.
services	List<DeviceService>	Indicates the device service list. For details, see **DeviceService structure**.

DeviceInfo structure

Parameter	Type	Description
nodeId	String(256)	Identifies a device.
name	String(256)	Indicates the device name.
description	String(2048)	Indicates the device description.
manufacturerId	String(256)	Uniquely identifies a manufacturer.
manufacturerName	String(256)	Indicates the manufacturer name.
mac	String(256)	Indicates the MAC address of the device.
location	String(2048)	Indicates the device location.

Parameter	Type	Description
deviceType	String(256)	Indicates the device type. The upper camel case is used, for example, **MultiSensor**, **ContactSensor**, and **CameraGateway**.
model	String(256)	Indicates the device model. In Z-Wave, the format is productType + productId. The value is a hexadecimal value in the format of XXXX-XXXX. Zeros are added if required, for example, **001A-0A12**. The format in other protocols is still to be determined.
swVersion	String(256)	Indicates the software version of the device. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	String(256)	Indicates the firmware version of the device.
hwVersion	String(256)	Indicates the hardware version of the device.
protocolType	String(256)	Indicates the protocol type used by the device. The value options are **CoAP**, **huaweiM2M**, **Z-Wave**, **ONVIF**, **WPS**, **Hue**, **WiFi**, **J808**, **Gateway**, **ZigBee**, and **LWM2M**.
bridgeId	String(256)	Identifies the bridge through which the device accesses the IoT platform.
status	String	Indicates whether the device is online. The value options are **ONLINE**, **OFFLINE**, **INBOX**, and **ABNORMAL**.
statusDetail	String(256)	Indicates status details of the device. The value of this parameter varies with the value of **status**. For details, see **status and statusDetail**.
mute	String	Indicates whether the device is in the frozen state. Based on the value of this parameter, the IoT platform determines whether to manage and store data reported by the device. ● **TRUE**: The device is in the frozen state. ● **FALSE**: The device is not in the frozen state.
supportedSecurity	String	Indicates whether the security mode is supported. ● **TRUE**: The security mode is supported. ● **FALSE**: The security mode is not supported.
isSecurity	String	Indicates whether the security mode is enabled. ● **TRUE**: The security mode is enabled. ● **FALSE**: The security mode is disabled.
signalStrength	String(256)	Indicates the signal strength of the device.

Parameter	Type	Description
sigVersion	String(256)	Indicates the SIG version of the device.
serialNumber	String(256)	Indicates the serial number of the device.
batteryLevel	String(256)	Indicates the battery level of the device.

status and statusDetail

status	statusDetail
OFFLINE	NONE CONFIGURATION_PENDING
ONLINE	NONE COMMUNICATION_ERROR CONFIGURATION_ERROR BRIDGE_OFFLINE FIRMWARE_UPDATING DUTY_CYCLE NOT_ACTIVE

NOTE

When the device status information is reported to the IoT platform, **status** and **statusDetail** must be included. It is recommended that **statusDetail** be used only for display but not for logical judgment.

DeviceService structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
serviceInfo	ServiceInfo	Indicates service information of the device. For details, see **ServiceInfo structure**.
data	ObjectNode(2097152)	Indicates an attribute value pair.
eventTime	String(256)	The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

ServiceInfo structure

Parameter	Type	Description
muteCmds	List<String>	Indicates the device command list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100218	The gatewayId and pageNo cannot be both null.	The **gatewayId** and **pageNo** parameters cannot be null at the same time. Recommended handling: Check whether **gatewayId** or **pageNo** is set.
400	100405	The request parameter is invalid.	The request message contains invalid parameters. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.8.3 Historical Device Data Query

Typical Scenario

The IoT platform receives and saves service data reported by devices during daily operation. The storage duration of device data can be configured by calling the API for modifying device information and the device data can be stored for a maximum of 90 days. If an NA needs to view the historical data reported by a device to the IoT platform, the NA can call this API to obtain the data.

API Function

This API is used by an NA to query historical data reported by a specified device to the IoT platform based on the device ID.

API Description

```
def queryDeviceDataHistory(self, qddhInDTO, accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qddhInDTO	Mandatory	QueryDeviceDataHistoryInDTO	query	For details, see QueryDeviceData-HistoryInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceDataHistoryInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Mandatory	String	query	Uniquely identifies a device.
gatewayId	Mandatory	String	query	Identifies a gateway.
serviceId	Optional	String	query	Identifies a service.
property	Optional	String	query	Indicates the service attribute.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.

Parameter	Mandatory or Optional	Type	Location	Description
startTime	Optional	String	query	Indicates the start time. Historical data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	query	Indicates the end time. Historical data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDataHistoryOutDTO

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
deviceDataHistoryDTOs	List<DeviceDataHistoryDTO>	Indicates a list of historical device data. For details, see **DeviceDataHistoryDTO structure**.

DeviceDataHistoryDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Uniquely identifies a device.
gatewayId	String(256)	Identifies a gateway.
appId	String(256)	Uniquely identifies an NA.
data	JsonObject	Indicates the data reported by the device.
timestamp	String(256)	Indicates the timestamp when the data is reported. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100418	The deviceData is not existed.	The device data does not exist. Recommended handling: ● If **deviceId** carried in the request is incorrect, check whether **deviceId** belongs to **appId** or whether **deviceId** is incorrect. ● Check whether **appId** carried in the header contains **deviceId**. ● If the URL contains the optional parameter **appId**, check whether the value of **appId** is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description. For example, check whether the value of **pageSize** exceeds 2000.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.8.4 Querying Historical Device Shadow Data

Typical Scenario

When an NA modifies the configuration of a device shadow by calling the API for modifying device shadow information, the IoT platform saves the modification record. If the NA needs to view historical configuration records of the device shadow, the NA can call this API to obtain the records.

API Function

This API is used by an NA to query historical configuration data about a device shadow based on the device ID.

API Description

```
def queryDeviceDesiredHistory(self, qddhInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qddhInDTO	Mandatory	QueryDeviceDesiredHistoryInDTO	query	For details, see QueryDeviceDesiredHistoryInDTO structure.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceDesiredHistoryInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Mandatory	String	query	Uniquely identifies a device.
gatewayId	Mandatory	String	query	Identifies a gateway.
serviceId	Optional	String	query	Identifies a service.
property	Optional	String	query	Indicates the service attribute.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to 0, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The default value is **1**.
startTime	Optional	String	query	Indicates the start time. Historical data generated later than the specified start time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
endTime	Optional	String	query	Indicates the end time. Historical data generated earlier than the specified end time is queried. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Response Parameters

QueryDeviceDesiredHistoryOutDTO structure

Parameter	Type	Description
totalCount	Long	Indicates the number of queried records.
pageNo	Long	Indicates the page number.
pageSize	Long	Indicates the number of records on each page.
DeviceDesiredHistoryDTO	List<DeviceDesiredHistoryDTO>	Indicates a list of historical device data. For details, see **DeviceDesiredHistoryDTO structure**.

DeviceDesiredHistoryDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
deviceId	String(256)	Uniquely identifies a device.
gatewayId	String(256)	Identifies a gateway.
appId	String(256)	Uniquely identifies an NA.
desired	JsonObject	Indicates the data reported by the device.
timestamp	String(256)	Indicates the timestamp when the data is configured. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
400	100216	The application input is invalid.	The application input is invalid. Recommended handling: Check whether parameters in the API request are correct by referring to the request parameter description.
400	100419	The deviceId and gatewayId can't be both null.	The **deviceId** and **gatewayId** parameters cannot be null at the same time. Recommended handling: Check whether **deviceId** or **gatewayId** is set.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.8.5 Querying Service Capabilities of a Device

Typical Scenario

If an NA needs to know which service attributes can be reported by a device and which commands can be delivered to the device, the NA can call this API to query the device service capabilities defined in the profile file of the device on the IoT platform.

API Function

This API is used by an NA to query device service capabilities, such as service attributes and device commands.

API Description

```
def queryDeviceCapabilities(self, qdcInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdcInDTO	Mandatory	QueryDeviceCapabilitiesInDTO	query	For details, see QueryDeviceCapabilitiesInDTO structure.

Parameter	Mandatory or Optional	Type	Location	Description
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceCapabilitiesInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
gatewayId	Optional	String	query	Identifies a gateway.
appId	Mandatory	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
deviceId	Optional	String	query	Uniquely identifies a device.

Response Parameters

QueryDeviceCapabilitiesOutDTO

Parameter	Type	Description
deviceCapabilities	List<DeviceCapabilityDTO>	Indicates the query result list. For details, see **DeviceCapabilityDTO structure**.

DeviceCapabilityDTO structure

Parameter	Type	Description
deviceId	String(256)	Uniquely identifies a device.
serviceCapabilities	List<ServiceCapabilityDTO>	Indicates the service capability list. For details, see **ServiceCapabilityDTO structure**.

ServiceCapabilityDTO structure

Parameter	Type	Description
serviceId	String(256)	Identifies a service.
serviceType	String(256)	Indicates the service type.
option	String(256)	Indicates a service option.
description	String(10240)	Indicates the service description.
commands	List<ServiceCommand>	Indicates the supported commands. For details, see **ServiceCommand structure**.
properties	List<ServiceProperty>	Indicates the attribute list. For details, see **ServiceProperty structure**.

ServiceCommand structure

Parameter	Type	Description
commandName	String(256)	Indicates the command name.
paras	List<ServiceCommandPara>	Indicates the attribute list. For details, see **ServiceCommandPara structure**.
responses	List<ServiceCommandResponse>	Indicates the response list. For details, see **ServiceCommandResponse structure**.

ServiceCommandPara structure

Parameter	Type	Description
paraName	String(256)	Indicates the parameter name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.
min	String	Indicates the minimum value of the attribute.
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

ServiceCommandResponse structure

Parameter	Type	Description
responseName	String(256)	Indicates the response name.
paras	List<ServiceCommandPara>	Indicates the attribute list. For details, see **ServiceCommandPara structure**.

ServiceProperty structure

Parameter	Type	Description
propertyName	String(256)	Indicates the attribute name.
dataType	String(256)	Indicates the data type.
required	Boolean	Indicates whether the parameter is mandatory.
min	String	Indicates the minimum value of the attribute.
max	String	Indicates the maximum value of the attribute.
step	Double	Indicates the step.
maxLength	Integer	Indicates the maximum length.
method	String(256)	Indicates the access method. The values are as follows: ● **R**: Readable ● **W**: Writable ● **E**: Observable
unit	String	Indicates the unit (symbol).
enumList	List<String>	Indicates the enumeration type list.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	100022	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9 Device Group Management

4.3.9.1 Creating a Device Group

Typical Scenario

An NA can call this API to create device groups on the IoT platform, and allocate devices to different device groups for group management. A device can be bound to multiple device groups.

When the NA needs to perform operations on devices (such as upgrading device software and firmware or delivering commands to devices in batches), the NA can select devices to be operated by device group.

API Function

This API is used by an NA to create device groups on the IoT platform to manage devices by group.

API Description

```
def createDeviceGroup(self, cdgInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cdgInDTO	Mandatory	CreateDeviceGroupInDTO	body	For details, see **CreateDeviceGroupInDTO structure**.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateDeviceGroupInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.

Parameter	Mandatory or Optional	Type	Location	Description
appId	Optional	String (50)	body	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
maxDevNum	Optional	Integer(>=0)	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.
deviceIds	Optional	List<String>	body	Identifies the devices to be added to the device group.
id	Optional	String(1-50)	Body	Identifies the device group.

Response Parameters

CreateDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group. The device group ID is automatically generated by the IoT platform.
appId	String(50)	Uniquely identifies an NA.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.
deviceIds	List<String>	Identifies the devices to be added to the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.
200	100607	The devGroup has reached the limit.	The number of device groups reaches the limit. Recommended handling: Check whether the number of created device groups reaches the upper limit specified in the license.
400	100609	Too much devices to add.	Too many devices are added to the device group. Recommended handling: Ensure that the number of device IDs contained in **deviceIds** is within the range specified by **maxDevNum**.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.2 Deleting a Device Group

Typical Scenario

If a device group is no longer needed on the IoT platform due to group changes, an NA can call this API to delete a specified device group.

API Function

This API is used by an NA to delete the configuration information about a device group by device group ID on the IoT platform.

API Description

```
def deleteDeviceGroup(self, devGroupId, accessAppId, accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed	The device group does not exist. Recommended handling: Check whether the device group ID is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.3 Modifying a Device Group

Typical Scenario

If information about a device group (such as the device group name and the device quantity limit in the device group) needs to be modified due to service changes, an NA can call this API to modify the information.

API Function

This API is used to modify the information of a specified device group on the IoT platform.

API Description

```
def modifyDeviceGroup(self, mdgInDTO, devGroupId, accessAppId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
mdgInDTO	Mandatory	ModifyDeviceGroupInDTO	body	For details, see **ModifyDeviceGroupInDTO structure**.

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String (50)	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

ModifyDeviceGroupInDTO

Parameter	Mandatory or Optional	Type	Location	Description
name	Mandatory	String(1-50)	body	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	Optional	String(1024)	body	Indicates the device group description.
maxDevNum	Optional	Integer(>=0)	body	Indicates the maximum number of devices in a device group. The default value is **0**. If the value is **0**, the number of devices is not limited.

Response Parameters

ModifyDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.

Parameter	Type	Description
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies the device group.
appId	String(50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group. If the value is **0**, the number of devices is not limited.
curDevNum	Integer	Indicates the current number of devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100602	The device group name has been used.	The device group name exists. Recommended handling: Change the device group name in the API request.

HTTP Status Code	Error Code	Error Description	Remarks
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.4 Querying Details About a Device Group

Typical Scenario

An NA can call this API to query information of all the created device groups to check the group details and usage of the device groups.

API Function

This API is used by an NA to query information about all created device groups on the IoT platform.

API Description

```
def queryDeviceGroups(self, qdgInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdgInDTO	Mandatory	QueryDeviceGroupsInDTO	query	For details, see **QueryDeviceGroupsInDTO**.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceGroupsInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
pageNo	Optional	Integer	query	Indicates the page number. ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of device group records on each page. The default value is **1**.
name	Optional	String	query	Indicates the device group name.

Response Parameters

QueryDeviceGroupsOutDTO structure

Parameter	Type	Description
totalCount	long	Indicates the total number of device groups.
pageNo	long	Indicates the page number.

Parameter	Type	Description
pageSize	long	Indicates the number of device group records on each page.
list	List<object>	Indicates the device group details.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.5 Querying Information About a Specified Device Group

Typical Scenario

An NA can call this API to query information about a specified device group to check the usage of the device group.

API Function

This API is used by an NA to query the information about a device group by device group ID on the IoT platform.

API Description

```
def querySingleDeviceGroup(self, devGroupId, accessAppId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	path	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QuerySingleDeviceGroupOutDTO structure

Parameter	Type	Description
name	String(50)	Indicates the device group name. The value can contain only uppercase and lowercase letters and digits.
description	String(1024)	Indicates the device group description.
id	String(50)	Identifies a device group.

Parameter	Type	Description
appId	String (50)	Identifies the application to which the device group belongs.
maxDevNum	Integer(>=0)	Indicates the maximum number of devices in the device group.
curDevNum	Integer	Indicates the current number of devices in the device group.
creator	String(1-50)	Indicates the name of the user who creates the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
200	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.6 Querying Members in a Specified Device Group

Typical Scenario

An NA can call this API to query information about members in a specified device group.

API Function

This API is used by an NA to query the information about a device in a specified device group on the IoT platform.

API Description

```
def queryDeviceGroupMembers(self, qdgmInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qdgmInDTO	Mandatory	QueryDeviceGroupMembersInDTO	query	For details, see **QueryDeviceGroupMembersInDTO structure**.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryDeviceGroupMembersInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String	query	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer(1000)	query	Indicates the number of devices on each page. The default value is **10**.

Response Parameters

QueryDeviceGroupMembersOutDTO structure

Parameter	Type	Description
totalCount	long	Indicates the number of devices in the device group.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of devices on each page.
deviceIds	List<String>	Identifies the devices in the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	107001	The serviceId is not exist.	The service ID does not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107002	The properties is empty in database.	The device attributes do not exist. Recommended handling: Check whether **serviceId** carried in the API request is correct.
400	107003	The request properties is unknown.	The device status is unknown. Recommended handling: Check whether the connection between the device and the IoT platform is normal.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.7 Adding Members to a Device Group

Typical Scenario

An NA can call this API to add a new device or an existing device to a specified device group. Before adding a device to a device group, you are advised to query the current number of devices and the maximum number of devices allowed in the device group by calling the API for querying information about a specified device group.

API Function

This API is used by an NA to add devices to a specified device group on the IoT platform.

API Description

```
def addDevicesToGroup(self, dgwdlDTO, accessAppId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dgwdlDTO	Mandatory	DeviceGroupWithDeviceListDTO	body	For details, see DeviceGroupWithDeviceListDTO structure.

Parameter	Mandatory or Optional	Type	Location	Description
accessAppId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeviceGroupWithDeviceListDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies the devices to be added to the device group.
accessAppId	Optional	String	query	If the device group belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.

Response Parameters

DeviceGroupWithDeviceListDTO

Parameter	Type	Description
devGroupId	String(1-50)	Identifies a device group.
deviceIds	List<String>	Identifies the devices to be added to the device group.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.9.8 Deleting Members from a Device Group

Typical Scenario

If one or more devices in a device group do not belong to the device group any longer, an NA can call this API to delete them from the device group.

API Function

This API is used by an NA to delete devices from a specified device group on the IoT platform.

API Description

```
def deleteDevicesFromGroup(self, dgwdlDTO, accessAppId, accessToken)
```


Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
dgwdlDTO	Mandatory	DeviceGroupWithDeviceListDTO	body	For details, see DeviceGroupWithDeviceListDTO structure.
accessAppId	Optional	String	query	If the device belongs to the current application, set this parameter to **None**. Otherwise, set this parameter to the ID of the authorized application.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

DeviceGroupWithDeviceListDTO

Parameter	Mandatory or Optional	Type	Location	Description
devGroupId	Mandatory	String(1-50)	body	Identifies a device group. The value of this parameter is returned by the IoT platform after the device group is added.
deviceIds	Mandatory	List<String>(1000)	body	Identifies devices to be deleted from the device group.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
200	100601	The number of device in the group has reach the max.	The number of devices in the device group reaches the upper limit. Recommended handling: Ensure that the number of devices in the device group is within the range specified by **maxDevNum**.
200	100603	The device group is not existed.	The device group does not exist. Recommended handling: Check whether the device group ID is correct.
400	100604	The device group request parameter is invalid.	The request message contains invalid parameters. Recommended handling: ● Check whether **deviceId** carried in the API request is correct. ● Check whether the number of devices in the device group reaches the upper limit.
400	50400	The input is invalid.	An input parameter is invalid. Recommended handling: Check whether parameters carried in the API call request are valid.
403	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
403	100217	The application hasn't been authorized.	The application has not been authorized. Recommended handling: In scenarios where applications are not authorized, ensure that request parameter **appId** is null.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
500	100203	The application is not existed.	The application does not exist. Recommended handling: ● Check whether **appId** carried in the HTTP request header is correct. ● Check whether **appId** in the request path (URL) is correct.
500	50252	Internal server error.	An internal server error occurs. Recommended handling: An internal error occurs on the IoT platform. Contact IoT platform maintenance personnel.

4.3.10 Device Upgrade

4.3.10.1 Querying a Version Package List

Typical Scenario

Before upgrading the device version, an NA can call this API to query the version upgrade packages that have been uploaded to the IoT platform to ensure that the target version package has been uploaded.

API Function

This API is used by an NA to query a list of uploaded version packages that meet a specified condition.

API Description

```
def queryUpgradePackageList(self, quplInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
quplInDTO	Mandatory	QueryUpgradePackageListInDTO	query	For details, see QueryUpgradePackageListInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradePackageListInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileType	Optional	String(256)	query	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	Optional	String(256)	query	Indicates the type of the device to which the version package is applicable.
model	Optional	String(256)	query	Indicates the model of the device to which the version package is applicable.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device to which the version package is applicable.
version	Optional	String(256)	query	Indicates the version of the version package.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.

Parameter	Mandatory or Optional	Type	Location	Description
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradePackageListOutDTO structure

Parameter	Type	Description
data	List< QueryUpgradePackageOutDTO>	Indicates the version package list. For details, see **QueryUpgradePackageOutDTO structure**.
pageNo	Integer	Indicates the page number.
pageSize	Integer	Indicates the number of records to be displayed on each page.
totalCount	Integer	Indicates the total number of query results.

QueryUpgradePackageOutDTO structure

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.

Parameter	Type	Description
description	String	Indicates the version package description.
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the values of **pageNo** and **pageSize** in the API request are within the valid ranges.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.

4.3.10.2 Querying a Specified Version Package

Typical Scenario

Before upgrading the device version, an NA can call this API to query the target version upgrade package to ensure that it has been uploaded to the IoT platform.

API Function

This API is used by an NA to query a specified version package based on the version package ID on the IoT platform. The version package ID can be obtained by calling the API for querying a version package list.

API Description

```
def queryUpgradePackage(self,fileId,accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QueryUpgradePackageOutDTO structure

Parameter	Type	Description
fileId	String	Identifies a version package.
name	String	Indicates the version package name.
version	String	Indicates the version of the version package.
fileType	String	Indicates the type of a version package. ● **firmwarePackage**: Indicates a firmware package. ● **softwarePackage**: Indicates a software package.
deviceType	String	Indicates the type of the device to which the version package is applicable.
model	String	Indicates the model of the device to which the version package is applicable.
manufacturerName	String	Indicates the manufacturer of the device to which the version package is applicable.

Parameter	Type	Description
protocolType	String	Indicates the protocol used by the device to which the version package is applicable.
description	String	Indicates the version package description.
date	String	Indicates the date on which the version package was generated.
uploadTime	String	Indicates the date on which the version package was uploaded.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

4.3.10.3 Deleting a Specified Version Package

Typical Scenario

If a device version package is no longer needed, an NA can call this API to delete the version package from the IoT platform.

API Function

This API is used by an NA to delete a specified version package from the IoT platform based on the version package ID. The ID of the version package to be deleted can be obtained by calling the API for querying a version package list.

API Description

```
def deleteUpgradePackage(self, fileId, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	path	Identifies a version package. The value of this parameter is obtained after the version package is uploaded.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

void

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

4.3.10.4 Creating a Software Upgrade Task

Typical Scenario

If the device software needs to be upgraded, an NA can call this API to create a software upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the software of multiple devices on the IoT platform. Currently, only the software of NB-IoT devices can be upgraded.

API Description

```
def createSoftwareUpgradeTask(self, cutInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cutInDTO	Mandatory	CreateUpgradeTaskInDTO	body	For details, see CreateUpgradeTaskInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateUpgradeTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	body	Identifies the target firmware version.
targets	Mandatory	OperateDevices	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.
policy	Optional	OperatePolicy	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	Optional	String	body	Specifies the type of a device. This parameter must be specified when a device group is specified.

Parameter	Mandatory or Optional	Type	Location	Description
model	Optional	String	body	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	Optional	String	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.

Parameter	Mandatory or Optional	Type	Location	Description
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies an operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.

HTTP Status Code	Error Code	Error Description	Remarks
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.

HTTP Status Code	Error Code	Error Description	Remarks
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

4.3.10.5 Creating a Firmware Upgrade Task

Typical Scenario

If the device firmware needs to be upgraded, an NA can call this API to create a firmware upgrade task for multiple devices. Before the upgrade, ensure that the target version package has been uploaded to the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Function

This API is used by an NA to upgrade the firmware of multiple devices on the IoT platform. Currently, only the firmware of NB-IoT devices can be upgraded.

API Description

```
def createFirmwareUpgradeTask(self, cutInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
cutInDTO	Mandatory	CreateUpgradeTaskInDTO	body	For details, see CreateUpgradeTaskInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

CreateUpgradeTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
fileId	Mandatory	String	body	Identifies the target firmware version.
targets	Mandatory	OperateDevices	body	Indicates the devices to be upgraded. For details, see **OperateDevices structure**.

Parameter	Mandatory or Optional	Type	Location	Description
policy	Optional	OperatePolicy	body	Indicates the execution policy of an upgrade task. For details, see **OperatePolicy structure**.

OperateDevices structure

Parameter	Mandatory or Optional	Type	Location	Description
deviceGroups	Optional	List<String>	body	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	Optional	String	body	Specifies the type of a device. This parameter must be specified when a device group is specified.
model	Optional	String	body	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	Optional	String	body	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	Optional	List<String>	body	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Mandatory or Optional	Type	Location	Description
executeType	Mandatory	String	body	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	Optional	String	body	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	Optional	String	body	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Optional	Boolean	body	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Optional	Integer	body	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

Response Parameters

CreateUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies an operation task.

Error Codes

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the parameters in the request are correct.
400	123016	The parameter is error, targetversion not match with device.	The value of **targetVersion** is incorrect. Specifically, it does not match the specified device. Recommended handling: Check whether the values of **deviceType**, **manufacturerName**, and **model** carried in the API request are consistent with the target version package information specified by **fileId**.
400	123019	manufacturerName is null.	The manufacturer name is null. Recommended handling: Check whether **manufacturerName** carried in the API request is correct.
400	123020	deviceType is null	The device type is null. Recommended handling: Check whether **deviceType** carried in the API request is correct.
400	123021	model is null.	The device model is null. Recommended handling: Check whether **model** carried in the API request is correct.
400	123022	deviceGroups and devices cannot be null together	**deviceGroups** and **devices** cannot be both null. Recommended handling: Specify either **deviceGroups** or **devices**.
400	123023	deviceGroups and devices cannot be exist together	**deviceGroups** and **devices** cannot coexist. Recommended handling: Specify either **deviceGroups** or **devices**.

HTTP Status Code	Error Code	Error Description	Remarks
400	123024	The number of deviceGroups or devices reached upper limit	The number of device groups or devices reaches the upper limit. Recommended handling: Check the number of device groups or devices. The number cannot exceed 256.
400	123025	executeType is error or can not to be null.	The value of **executeType** is incorrect or is not specified. Recommended handling: Check whether **executeType** in the request is unspecified or incorrect.
400	123026	startTime or endTime is null or error.	The value of **startTime** or **endTime** is empty or incorrect. Recommended handling: Check whether **startTime** and **endTime** in the request are unspecified or incorrect.
400	123028	retryTimes is null or beyond the limit.	The value of **retryTimes** is empty or exceeds the upper limit. Recommended handling: Verify that the value of **retryTimes** in the request is left unspecified or is not less than 1 or greater than 5.
400	123032	startTime can not be later than the endTime.	The value of **startTime** cannot be later than the value of **endTime**. Recommended handling: Check whether **startTime** in the request is later than **endTime**.
400	123033	startTime can not be earlier than the now.	The value of **startTime** cannot be earlier than the current time. Recommended handling: Check whether **startTime** in the request is earlier than the current time.

HTTP Status Code	Error Code	Error Description	Remarks
400	123034	endtime must be greater than 5 minutes.	The value of **endtime** must be 5 minutes later than that of **startTime**. Handling suggestions: Verify that the interval between **startTime** and **endTime** in the request is greater than 5 minutes.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123002	Device or package file not found.	The device or package does not exist. Recommended handling: Check whether **fileId** carried in the API request is correct.

4.3.10.6 Querying the Result of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, an NA can call this API to query details about the upgrade task, including the configuration and execution status.

API Function

This API is used by an NA to query details about a software or firmware upgrade task, including the configuration and execution status.

API Description

```
def queryUpgradeTask(self, operationId,  accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

Response Parameters

QueryUpgradeTaskOutDTO structure

Parameter	Type	Description
operationId	String	Identifies an operation task.
createTime	String	Indicates the time when the operation task was created.
startTime	String	Indicates the time when the operation task was started.
stopTime	String	Indicates the time when the operation task was stopped.
operateType	String	Indicates the operation type. ● firmware_upgrade ● software_upgrade
targets	OperateDevices	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
policy	OperatePolicy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.

Parameter	Type	Description
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	OperationStaResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
extendPara	JsonString	Indicates an extended parameter of the operation task. For details, see **extendPara request parameter**.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Specifies the type of a device. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.

Parameter	Type	Description
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

extendPara request parameter when **operateType** is set to **softwareUpgrade** or **firmwareUpgrade**

Parameter	Type	Description
fileVersion	String	Identifies the target version.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

4.3.10.7 Querying Details About Subtasks of a Specified Upgrade Task

Typical Scenario

After a device software or firmware upgrade task is created, the upgrade of each device involved in the task is a subtask (the number of subtasks is the same as that of the devices

involved in the task). An NA can call this API to query details about subtasks of the upgrade task to check their execution status.

API Function

This API is used by an NA to query upgrade status of each device involved in a software or firmware upgrade task.

API Description

```
def queryUpgradeSubTask(self, qustInDTO, operationId,  accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qustInDTO	Mandatory	QueryUpgradeSubTaskInDTO	query	For details, see QueryUpgradeSubTaskInDTO structure.
operationId	Mandatory	String	path	Identifies an operation task. The value of this parameter is returned by the IoT platform after the operation task is created.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradeSubTaskInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
subOperationStatus	Optional	String	query	Indicates subtask status. If this parameter is not specified, execution details of all subtasks under the operation task are queried. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeSubTaskOutDTO structure

Parameter	Type	Description
data	list<SubOperationInfo>	Indicates the subtask list information. For details, see **SubOperationInfo**.
pageNo	long	Indicates the page number.
pageSize	long	Indicates the number of records to be displayed on each page.
totalCount	long	Indicates the total number of query results.

SubOperationInfo structure

Parameter	Type	Description
subOperationId	String	Identifies a subtask.
createTime	String	Indicates the time when the subtask was created.
startTime	String	Indicates the time when the subtask was started.
stopTime	String	Indicates the time when the subtask was stopped.
operateType	String	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
deviceId	String	Identifies a device on which the subtask is performed.
status	String	Indicates the subtask status. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **fail**: The subtask fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
detailInfo	String	Indicates detailed description of the subtask status. In case of a failure, the value of this parameter is the failure cause.
extendInfo	JsonString	Indicates extended information of the subtask. The value of this parameter varies depending on the operation type.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.

HTTP Status Code	Error Code	Error Description	Remarks
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

4.3.10.8 Querying an Upgrade Task List

Typical Scenario

An NA can call this API to query the created upgrade tasks to view the detailed information and execution status of each upgrade task.

API Function

This API is used by an NA to query details about upgrade tasks that meet specified conditions.

API Description

```
def queryUpgradeTaskList(self, qutlInDTO, accessToken)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Location	Description
qutlInDTO	Mandatory	QueryUpgradeTaskListInDTO	query	For details, see QueryUpgradeTaskListInDTO structure.
accessToken	Mandatory	String	header	This parameter is set to the value of the access token obtained by calling the **Authentication** API.

QueryUpgradeTaskListInDTO structure

Parameter	Mandatory or Optional	Type	Location	Description
operationType	Optional	String(256)	query	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
operationStatus	Optional	String(256)	query	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
deviceType	Optional	String(256)	query	Indicates the device type for which the operation task is intended.
model	Optional	String(256)	query	Indicates the device model for which the operation task is intended.
manufacturerName	Optional	String(256)	query	Indicates the manufacturer of the device for which the operation task is intended.
deviceId	Optional	String(256)	query	Identifies the device for which the operation task is intended.

Parameter	Mandatory or Optional	Type	Location	Description
pageNo	Optional	Integer	query	Indicates the page number. Default value: **0** ● If the value is **null**, pagination query is not performed. ● If the value is an integer greater than or equal to **0**, pagination query is performed. ● If the value is **0**, the first page is queried.
pageSize	Optional	Integer	query	Indicates the number of records on each page. The value ranges from 1 to 100. The default value is **10**.

Response Parameters

QueryUpgradeTaskListOutDTO structure

Parameter	Type	Description
data	List<OperationInfo>	Indicates the task list. For details, see **OperationInfo structure**.
pageNo	Integer	Indicates the page number.
pageSize	Integer	Indicates the number of records to be displayed on each page.
totalCount	Integer	Indicates the total number of query results.

OperationInfo structure

Parameter	Type	Description
operationId	String	Identifies an operation task.
createTime	String	Indicates the time when the operation task was created.
startTime	String	Indicates the time when the operation task was started.
stopTime	String	Indicates the time when the operation task was stopped.

Parameter	Type	Description
operateType	String	Indicates the operation type. ● **firmware_upgrade** ● **software_upgrade**
targets	OperateDevices	Indicates the target devices on which the operation task is performed. For details, see **OperateDevices structure**.
policy	OperatePolicy	Indicates the execution policy of the operation task. For details, see **OperatePolicy structure**.
status	String	Indicates the status of the operation task. ● **wait**: The operation task is waiting to be executed. ● **processing**: The operation task is being executed. ● **failed**: The operation task fails to be executed. ● **success**: The operation task is successfully executed. ● **stop**: The operation task stops being executed.
staResult	OperationStaResult	Indicates operation result statistics. For details, see **OperationStaResult structure**.
extendPara	JsonString	Indicates extended information of the operation task. The value of this parameter varies depending on the operation type.

OperateDevices structure

Parameter	Type	Description
deviceGroups	List<String>	Indicates the device group name list. A maximum of 256 device groups are supported. Either this parameter or **devices** must be specified.
deviceType	String	Specifies the type of a device. This parameter must be specified when a device group is specified.
model	String	Indicates the device model. This parameter must be specified when a device group is specified.

Parameter	Type	Description
manufacturerName	String	Indicates the manufacturer name. This parameter must be specified when a device group is specified.
devices	List<String>	Indicates the device ID list. A maximum of 256 devices are supported. Either this parameter or **deviceGroups** must be specified.

OperatePolicy structure

Parameter	Type	Description
executeType	String	Indicates the execution type. The default value is **now**. ● **now**: The upgrade task is executed now. ● **device_online**: The upgrade task is executed when a device goes online. ● **custom**: The execution type is customized.
startTime	String	Indicates the start time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
endTime	String	Indicates the end time of the operation task. This parameter must be specified when **executeType** is set to **custom**. The value is in the format of yyyyMMdd'T'HHmmss'Z'. An example value is **20151212T121212Z**.
retryType	Boolean	Indicates whether the platform retries the task upon an execution failure. The default value is **false**. ● **true**: Retry is performed. ● **false**: The platform does not retry the task.
retryTimes	Integer	Indicates the number of retries. The value ranges from 1 to 5. This parameter must be specified when **retryType** is set to **true**.

OperationStaResult structure

Parameter	Type	Description
total	Integer(64)	Indicates the number of operated devices.
wait	Integer(64)	Indicates the number of devices waiting to be operated.
processing	Integer(64)	Indicates the number of devices that are being operated.
success	Integer(64)	Indicates the number of devices that are operated successfully.
fail	Integer(64)	Indicates the number of devices that fail to be operated.
stop	Integer(64)	Indicates the number of devices that stop being operated.
timeout	Integer(64)	Indicates the number of devices that fail to be operated due to a timeout.

Error Code

HTTP Status Code	Error Code	Error Description	Remarks
400	120015	Bad request error.	A request error occurs. Recommended handling: Check whether the format of **fileId** carried in the API request is correct.
400	123029	pageNo or pageSize beyond the limit.	The value of **pageNo** or **pageSize** exceeds the upper limit. Recommended handling: Change the value of **pageNo** or **pageSize** to a valid value.
403	1010009	app throttle exceed.	The NA calls the API at a frequency that exceeds the flow control threshold (100 calls per minute by default). Recommended handling: Contact IoT platform maintenance personnel to adjust the flow control threshold or control the API call frequency.

HTTP Status Code	Error Code	Error Description	Remarks
403	1010005	App_key or access_token is invalid.	The access token is invalid. Recommended handling: Check whether **accessToken** carried in the API request is correct.
404	123009	The requested task does not exist.	The queried task does not exist. Recommended handling: Check whether **operationId** carried in the API request is correct.

5 AgentLite API Reference (Android)

5.1 Before You Start

1. Overview

The IoT AgentLite (AgentLite for short) provides standard capabilities for intelligent devices to access the IoT platform in smart home, industrial IoT, connected vehicles, and other fields. These intelligent devices, such as IP cameras (IPCs), lightweight gateways, industrial gateways, and head units, have strong computing capabilities.

2. API Overview

The following table lists the functions provided by the AgentLite, the APIs for implementing these functions, and the relationship between the APIs. It helps you quickly locate APIs that are used to develop specific services.

Function	API	Description
Access of directly connected devices	BaseService.init	Initializes AgentLite resources.
	BaseService.destroy	Releases AgentLite resources.
	BindConfig.setConfig	Configures device binding.
	BindService.bind	Binds devices.
	HubService.TOPIC_UNBINDDEVICE	Receives device unbinding commands.

Function	API	Description
	LoginConfig.setConfig	Configures login information.
	LoginService.login	Connects devices to the IoT platform.
	LoginService.logout	Disconnects devices from the IoT platform.
Management of non-directly connected devices on a gateway	HubService.addDevice	Adds devices.
	HubService.deviceStatusUpdate	Updates device status.
	HubService.rmvDevice	Deletes devices.
Reporting device service data	DataTransService.reportData	Reports device service data.
Receiving device service commands	DataTransService.TOPIC_COMMAND_RECEIVE	Receives device service commands.

5.2 APIs

The external APIs provided by the AgentLite can be used for broadcasting, access of directly connected devices, management of non-directly connected devices, reporting device service data, and receiving device service commands.

5.2.1 Broadcasting

The local broadcast of the Android system is used to receive messages reported by the AgentLite.

Therefore, broadcasts can be registered only in the dynamic registration mode.

5.2.2 Access of Directly Connected Devices

After obtaining the AgentLite, a third-party developer can connect the device to the IoT platform.

- Directly connected devices: Devices are directly connected to the IoT platform through device binding and login.
- Non-directly connected devices: Devices are connected to the IoT platform through gateways.

5.2.2.1 Initializing AgentLite Resources

API Function

This API is used to initialize AgentLite resources.

API Description

```
public static boolean init(String workPath, String logPath, Context context);
```

Class

BaseService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
workPath	Mandatory	String	Specifies the AgentLite working path, which is used to store the AgentLite configuration files and temporary files. The working path must be valid.
logPath	Optional	String	Specifies the log path. (If this parameter is left unspecified, logs are written to the working path.)
context	Mandatory	Context	Specifies the Android application context.

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Call this API to initialize AgentLite resources.
BaseService.init("/sdcard/helloWorld", null, context);
```

5.2.2.2 Releasing AgentLite Resources

API Function

This API is used to release all dynamic resources that are applied for, such as the memory and thread.

API Description

```
public static void destroy();
```

Class

BaseService

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Call this API to release AgentLite resources.
BaseService.destroy();
```

5.2.2.3 Binding Configuration

API Function

This API is used to configure the IP address and port number of the IoT platform before binding a device.

API Description

```
public static boolean setConfig(int key, String value);
```

Class

BindConfig

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	int	Specifies the configuration items of device binding. ● IP address: BindConfig.BIND_CONFIG_ADDR ● Port number: BindConfig.BIND_CONFIG_PORT
value	Mandatory	String	Specifies the value of the configuration item. ● IP address: IP address of the IoT platform to which the AgentLite is connected ● Port number: 8943

Return Value

Return Value	Description
true	Success
false	Failure

Output

N/A

Example

```
BindConfig.setConfig(BindConfig.BIND_CONFIG_ADDR, "127.0.0.1");
BindConfig.setConfig(BindConfig.BIND_CONFIG_PORT, "8943");
```

5.2.2.4 Binding a Device

API Function

A device must be bound to the IoT platform before accessing the IoT platform for the first time. The upper-layer application calls this API to transfer the device serial number, MAC address, or other device information to bind a device to the IoT platform.

Before binding a device, developers must call the **Binding Configuration** API to set the IP address and port number of the IoCM server to be bound. The default port number is 8943 for the AgentLite.

NOTE

Before a directly connected device accesses the IoT platform for the first time, developers must register the device with the IoT platform and then initiate a binding request on the device. If the device is not registered with the IoT platform, the binding fails. The AgentLite waits for a while and tries again.

API Description

```
public static boolean bind(String verifyCode, IotaDeviceInfo deviceInfo);
```

Class

BindService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
verifyCode	Mandatory	String	Specifies a device binding verification code. If the device is registered on the SP portal, set **verifyCode** to the preSecret used during device registration.
deviceInfo	Mandatory	**IotaDeviceInfo**	Specifies information about the device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

- Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the binding is successful. The binding is successful only after the **BindService.TOPIC_BINDDEVICE_RSP** broadcast is received.
- If the binding fails, the AgentLite automatically binds the device after 30 seconds. If the retry fails for five consecutive times (the total number of attempts is six), a message is returned indicating that the binding fails and the binding stops. If developers want to re-initiate the binding, restart the device.

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_BINDDEVICE_RSP	IotaMessage object (Obtained by using **intent.getSerializableExtra(BindService.BIND_BROADCAST_MSG_IE_IOTAMSG)**)	BIND_IE_RESULT	Specifies the binding result.
		BIND_IE_DEVICEID	Specifies the logical device ID assigned by the IoT platform.
		BIND_IE_DEVICESECRET	Specifies the authentication secret for a device to access the IoT platform.

Broadcast Name	Broadcast Parameter	Member	Description
		BIND_IE_APPID	Identifies an application.
		BIND_IE_HA_ADDR	Specifies the IP address of the HA server.
		BIND_IE_LVS_ADDR	Specifies the IP address of the LVS server.

Example

Call this API to bind a device.

```
BindService.bind(new IotaDeviceInfo("nodeId", "manufacturerId", " Gateway",
"model", "protocolType"));
```

Receive a device binding response.

```
//After a device is bound, the AgentLite will return the parameters shown in the
following output. Developers must store these parameters and configure these
parameters on the device before connecting the device to the IoT platform.
BroadcastReceiver mBindRsp;
mBindRsp = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
    //Do Something
    IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(BindService.BIND_BROADCAST_MSG_IE_IOTAMSG
);
    int result = iotaMsg.getUint(BindService.BIND_IE_RESULT, 0);
    String deviceId = iotaMsg.getString(BindService.BIND_IE_DEVICEID);
    String Secret = iotaMsg.getString(BindService.BIND_IE_DEVICESECRET);
    String Appid = iotaMsg.getString(BindService.BIND_IE_APPID);
    String haAddr = iotaMsg.getString(BindService.BIND_IE_HA_ADDR);
    String lvsAddr = iotaMsg.getString(BindService.BIND_IE_LVS_ADDR);
    return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterBind = new IntentFilter(BindService.TOPIC_BINDDEVICE_RSP);
mLocalBroadcastManager.registerReceiver(mBindRsp, filterBind);
```

5.2.2.5 Unbinding a Device

API Function

This API is used to register the broadcast for receiving the command used to unbind a directly connected device from the IoT platform. After this broadcast is received, developers must delete the configuration information about the directly connected device and release all resources. When the device restarts, developers must bind the device to the IoT platform again.

API Description

```
HubService.TOPIC_UNBINDDEVICE;
```

Class

HubService

Example

```
BroadcastReceiver mUnbindRsp;
mUnbindRsp = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        //Delete config file, free resource
        return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterUnbind = new IntentFilter(HubService.TOPIC_UNBINDDEVICE);
mLocalBroadcastManager.registerReceiver(mUnbindRsp, filterUnbind);
```

5.2.2.6 Configuring Parameters

5.2.2.6.1 Setting Service Parameters

API Function

This API is used to set the parameters required for device login.

API Description

```
public static boolean setConfig(int key, String value);
```

Class

LoginConfig

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	int	Specifies the configuration items for device login. ● Device ID: LoginConfig.LOGIN_CONFIG_DEVICEID ● Application ID: LoginConfig.LOGIN_CONFIG_APPID ● Password: LoginConfig.LOGIN_CONFIG_SECRET ● HTTP address: LoginConfig.LOGIN_CONFIG_IOCM_ADDR ● HTTP port: LoginConfig.LOGIN_CONFIG_IOCM_PORT ● MQTT address: LoginConfig.LOGIN_CONFIG_MQTT_ADDR ● MQTT port: LoginConfig.LOGIN_CONFIG_MQTT_PORT
value	Mandatory	String	Specifies the values of the configuration items. ● Device ID: obtained from the broadcast that indicates the device is bound successfully. ● Application ID: obtained from the broadcast that indicates the device is bound successfully. ● Secret: obtained from the broadcast that indicates the device is bound successfully. ● HTTP address: address for the AgentLite to interwork with the IoT platform ● HTTP port: 8943 ● MQTT address: address for the AgentLite to interwork with the IoT platform ● MQTT port: 8883

Return Value

Return Value	Description
true	Success
false	Failure

Output

N/A

Example

Save the parameters carried in the device binding response.

```
private void saveBindPara(IotaMessage iotaMsg) {
    LogUtil.i(this, TAG, "saveBindParaAndGotoLogin");
    String appId = iotaMsg.getString(BindService.BIND_IE_APPID);
    String deviceId = iotaMsg.getString(BindService.BIND_IE_DEVICEID);
    String secret = iotaMsg.getString(BindService.BIND_IE_DEVICESECRET);
    String haAddress = AgentLiteUtil.get(ConfigName.platformIP);

    saveGatewayInfo(appId, deviceId, secret, haAddress, null);
}
```

Set the parameters for device login.

```
private void configLoginPara() {
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_DEVICEID,
GatewayInfo.getDeviceID());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_APPID,
GatewayInfo.getAppID());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_SECRET,
GatewayInfo.getSecret());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_ADDR,
GatewayInfo.getHaAddress());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_PORT, "8943");
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_ADDR,
GatewayInfo.getHaAddress());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_PORT, "8883");
}
```

5.2.2.6.2 (Optional) Selecting an Encryption Algorithm Type

API Function

This API is used to configure the encryption algorithm type for sensitive information before login. Developers can select an appropriate encryption algorithm type based on the security level for their services.

API Description

```
public static boolean setAlgType (int type);
```

Class

BaseService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
type	Mandatory	int	Specifies the encryption algorithm type. The value ranges from 0 to 2. ● **0**: AES 256 CBC ● **1**: AES 128 GCM ● **2**: AES 256 GCM

Return Value

Return Value	Description
true	Success
false	Failure

Output

N/A

Example

Select an encryption algorithm type.

```
private void configAlgPara() {
    BaseService. setAlgType (2);
}
```

5.2.2.7 Connecting a Device

API Function

This API is used to connect a device to the IoT platform after the device is bound to the IoT platform for the first time or the device restarts.

API Description

```
public static boolean login();
```

Class

LoginService

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the login is successful. The login is successful only after the **LoginService.TOPIC_LOGIN_CONNECTED** broadcast is received. Before login, the API for **Setting Service Parameters** is used to transfer the required login information.

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_LOGIN_CONNECTED	IotaMessage object (Obtained by **using intent.getSerializableExtra(LoginService.LOGIN_BROADCAST_MSG_IE_IOTAMSG)**)	N/A	Specifies that the login or reconnection is successful.
TOPIC_LOGIN_DISCONNECT	IotaMessage object (Obtained by using **intent.getSerializableExtra(LoginService.LOGIN_BROADCAST_MSG_IE_IOTAMSG)**)	LOGIN_IE_REASON	Specifies the cause of the login or reconnection failure.

Example

```
//Set login parameters.
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_DEVICEID, "deviceId");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_APPID, "appId");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_SECRET, "passWord");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_ADDR, "haAddr");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_PORT, "8943");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_ADDR, "haAddr");
LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_PORT, "8883");

//Call this API to connect a device to the IoT platform.
LoginService.login();
```

The device then waits for a connection status broadcast from the AgentLite.

Suggestions:

1. If the connection is successful, report the status of the indirectly connected devices and the cached data.
2. If the connection fails, record the device status. If an indirectly connected device reports data, the data is cached and reported until the connection is successful.

```
//Receive a login success response.
BroadcastReceiver mReceiverConnect;
mReceiverConnect = new BroadcastReceiver() {
    @Override
       public void onReceive(Context context, Intent intent) {
    //Obtain IotaMessage.
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(LoginService.
LOGIN_BROADCAST_MSG_IE_IOTAMSG);
//Obtain the system status from the message handle returned by the callback.
        int status = uspMsg.getint(LoginService.LOGIN_IE_STATUS, 0);
        //update device states
        ...
        return true;
    }
}
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterCon= new IntentFilter(LoginService.TOPIC_LOGIN_CONNECTED);
mLocalBroadcastManager.registerReceiver(mReceiverConnect, filterCon);
//Receive a login failure response.
BroadcastReceiver mReceiverDisconnect;
mReceiverDisconnect = new BroadcastReceiver() {
    @Override
       public void onReceive(Context context, Intent intent) {
    //Obtain IotaMessage.
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(LoginService.
LOGIN_BROADCAST_MSG_IE_IOTAMSG);
//Obtain the error code of the response.
        int reason = iotaMsg.getint(LoginService.LOGIN_IE_REASON, 0);
        //stop reporting data
        ...
        return true;
    }
}
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterDiscon= new
IntentFilter(LoginService.TOPIC_LOGIN_DISCONNECTED);
mLocalBroadcastManager.registerReceiver(mReceiverDisconnect, filterDiscon);
```

After the login is successful, the device is connected to the IoT platform.

If the device is disconnected from the IoT platform due to the network or server fault, the AgentLite will automatically attempt to reconnect the device to the IoT platform and report the real-time status to the third-party application by using these two broadcasts.

5.2.2.8 Disconnecting a Device

API Function

This API is used to disconnect a device from the IoT platform.

API Description

```
public static boolean logout();
```

Class

LoginService

Parameter Description

N/A

Return Value

Return Value	Description
true	Success
false	Failure

Output

N/A

Example

```
LoginService.logout();
```

5.2.3 Management of Non-Directly Connected Devices on a Gateway

If the device on which the AgentLite is integrated is a gateway, it needs to manage all non-directly connected devices (sensors), including access and deletion of these devices. In addition, the gateway needs to record the mapping between logical and physical IDs of these devices.

5.2.3.1 Adding a Device

API Function

This API is used to add a device that accesses the IoT platform through the gateway. After the device is connected to the IoT platform, the IoT platform will assign a unique logical ID to the device.

API Description

```
public static boolean addDevice(int cookie, IotaDeviceInfo deviceInfo);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
deviceInfo	Mandatory	**IotaDeviceInfo** class	Specifies information about the device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the device is added successfully. The device is added successfully only after the **TOPIC_ADDDEV_RSP** broadcast is received.

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_ADDDEV_RSP	IotaMessage (Obtained by using **intent.getSerializableExtra(HubService.HUB_BROADCAST_IE_IOTAMSG)**)	HUB_IE_RESULT	Specifies the device adding result.
		HUB_IE_DEVICEID	Specifies the device ID. If the device is added successfully, the device ID is returned.
		HUB_IE_COOKIE	The value ranges from 1 to 65535.

Example

```
//Call this API to add a device.
HubService.addDevice(29011, new IotaDeviceInfo("nodeId", "manufacturerId",
"deviceType", "model", "protocolType"));
```

The device waits for the result.

```
//java code
//Register a broadcast receiver to process the device adding.
BroadcastReceiver mAdddeviceRsp;
mAdddeviceRsp = new BroadcastReceiver() {
    @Override
        public void onReceive(Context context, Intent intent) {
        //Do Something
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(HubService. HUB_BROADCAST_IE_IOTAMSG);
        int result = iotaMsg.getUint(HubService.HUB_IE_RESULT, 0);
        String deviceId = iotaMsg.getString(HubService.HUB_IE_DEVICEID);
        int cookie = iotaMsg.getUint(HubService.HUB_IE_COOKIE, 0);
        return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterAddDev = new IntentFilter(HubService.TOPIC_ADDDEV_RSP);
mLocalBroadcastManager.registerReceiver(mAdddeviceRsp, filterAddDev);
```


5.2.3.2 Updating Device Status

API Function

This API is used to update status information about devices, including directly connected devices and indirectly connected devices managed by the IoT platform. The device status can be updated through this API when the device is offline or online.

The status of directly connected devices is updated based on the device login status. If a directly connected device is disconnected from the IoT platform, its status becomes offline. If a directly connected device is connected or reconnected to the IoT platform, its status becomes online and does not need to be updated through this API. Therefore, it is recommended that developers call this API to update the status of indirectly connected devices.

API Description

```
public static boolean updateDeviceStatus(int cookie, String deviceId, String
status, String statusDetail);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
deviceId	Mandatory	String	Identifies a device.
status	Mandatory	String	Specifies the device status. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline.
statusDetail	Mandatory	String	Provides the detailed device status information. ● NONE ● CONFIGURATION_PENDING ● COMMUNICATION_ERROR ● CONFIGURATION_ERROR ● BRIDGE_OFFLINE ● FIRMWARE_UPDATING ● DUTY_CYCLE ● NOT_ACTIVE

Return Value

Return Value	Description
true	Success
false	Failure

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_DEVSTATUS_UPDATA_RSP	IotaMessage object (Obtained by using **intent.getSerializableExtra(HubService.HUB_BROADCAST_IE_IOTAMSG)**)	HUB_IE_RESULT	Specifies the device status update result.
		HUB_IE_DEVICEID	Identifies a device.

Example

```
HubService.deviceStatusUpdate(0, deviceId, "ONLINE", "NONE");
```

The device waits for the command execution result.

```
//Register a broadcast receiver to process the device status update.
BroadcastReceiver mReceiverDevStatus;
mReceiverDevStatus = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
   //Obtain IotaMessage.
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(HubService.HUB_BROADCAST_IE_IOTAMSG);
//Obtain the error code of the response.
        String result = iotaMsg.getString(HubService.HUB_IE_RESULT);
        String deviceId = iotaMsg.getString(HubService.HUB_IE_DEVICEID);
        ...
        return true;
    }
}
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterDiscon= new IntentFilter(HubService.
TOPIC_DEVSTATUS_UPDATA_RSP);
mLocalBroadcastManager.registerReceiver(mReceiverDevStatus, filterDiscon);
```

5.2.3.3 Deleting a Device

API Function

This API is used to delete indirectly connected devices from the IoT platform when a new device needs to be removed from the gateway.

API Description

```
public static boolean rmvDevice(int cookie, String deviceId);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
deviceId	Mandatory	String	Identifies a device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the device is deleted successfully. The device is deleted successfully only after the **TOPIC_RMVDEV_RSP** broadcast is received.

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_ RMVDEV_RSP	IotaMessage (Obtained by using **intent.getSerializableExtra(HubService.HUB_BROADCAST_IE_IOTAMSG)**)	HUB_IE_RESULT	Specifies the device deleting result.
		HUB_IE_COOKIE	The value ranges from 1 to 65535.

Example

Call this API to delete a device.

```
HubService.rmvDevice(122, deviceId);
```

The device waits for the result.


```
//java code
BroadcastReceiver mRmvdeviceRsp;
mRmvdeviceRsp = new BroadcastReceiver() {
    @Override
     public void onReceive(Context context, Intent intent) {
        //Do Something
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(HubService. HUB_BROADCAST_IE_IOTAMSG);
        int result = iotaMsg.getUint(HubService.HUB_IE_RESULT, 0);
        int cookie = iotaMsg.getUint(HubService.HUB_IE_COOKIE, 0);
        return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterRmvDev = new IntentFilter(HubService.TOPIC_RMVDEV_RSP);
mLocalBroadcastManager.registerReceiver(mDeldeviceRsp, filterRmvDev);
```

5.2.4 Reporting Device Service Data

API Function

This API is used by the gateway to report data of directly connected devices or indirectly connected devices to the IoT platform.

API Description

```
public static boolean dataReport(int cookie, String requstId, String deviceId,
String serviceId, String serviceProperties);
```

Class

DataTransService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
requstId	Mandatory	String	Identifies a request. This parameter is used to match the service command delivered by the IoT platform, which can be obtained from the broadcast described in **Receiving a Command**. ● Active data reporting: The value of this parameter is **NULL**. ● Command result reporting: The value of this parameter is the request ID of the command which the reported data matches.
deviceId	Mandatory	String	Identifies a device.
serviceId	Mandatory	String	Identifies a service.

Parameter	Mandatory or Optional	Type	Description
serviceProperties	Mandatory	String	Specifies the service attribute.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the service data is reported successfully. The service data is reported successfully only after the **TOPIC_DATA_REPORT_RSP** broadcast is received.

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_DATA_REPORT_RSP	IotaMessage (Obtained by using **intent.getSerializableExtra(DataTransService.DATATRANS_BROADCAST_IE_IOTAMSG)**)	DATATRANS_IE_RESULT	Specifies the data reporting result.
		DATATRANS_IE_COOKIE	The value ranges from 1 to 65535.

Example

A user uses JSON components to assemble service attributes (serviceProperties) based on the profile format.

```
DataTransService.dataReport(1211, NULL, "xxxx_xxxx_xxxx_xxxx", "DoorWindow",
"{\"status\":\"OPEN\"}");
```

The device waits for the data reporting result.

```
//Register a broadcast receiver to process the device service data reporting.
BroadcastReceiver mReportDataRsp;
mReportDataRsp = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        //Do Something
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(DataTransService.
DATATRANS_BROADCAST_IE_IOTAMSG);
```



```
        int cookie = iotaMsg.getUint(DataTransService.DATATRANS_IE_COOKIE, 0);
        int ret = iotaMsg.getUint(DataTransService.DATATRANS_IE_RESULT, 0);
        return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterReportData
= new IntentFilter(DataTransService.TOPIC_DATA_REPORT_RSP);
mLocalBroadcastManager.registerReceiver(mReportDataRsp, filterReportData);
```

5.2.5 Receiving a Command

API Function

This API is used to register the broadcast for receiving control commands from the IoT platform.

API Description

```
DataTransService.TOPIC_COMMAND_RECEIVE;
```

Parameter Description

N/A

Output

Broadcast Name	Broadcast Parameter	Member	Description
TOPIC_COMMAND_RECEIVE	IotaMessage (Obtained by using **intent.getSerializableExtra(DataTransService.DATATRANS_BROADCAST_IE_IOTAMSG)**)	DATATRANS_IE_REQUESTID	Identifies a request.
		DATATRANS_IE_DEVICEID	Specifies the logical ID of a device, which is allocated by the IoT platform when the device is added.
		DATATRANS_IE_SERVICEID	Identifies a service.
		DATATRANS_IE_METHOD	Specifies the service method.
		DATATRANS_IE_CMDCONTENT	Specifies the command content.

Example

Register a broadcast receiver to process the command reception.

```
BroadcastReceiver mReceiveCmd;
mReceiveCmd = new BroadcastReceiver() {
    @Override
```

```
    public void onReceive(Context context, Intent intent) {
        //Do Something
        IotaMessage iotaMsg =
(IotaMessage)intent.getSerializableExtra(DataTransService.
DATATRANS_BROADCAST_IE_IOTAMSG);
        String requstId =
iotaMsg.getString(DataTransService.DATATRANS_IE_REQUESTID);
        String deviceId =
iotaMsg.getString(DataTransService.DATATRANS_IE_DEVICEID);
        String serviceId =
iotaMsg.getString(DataTransService.DATATRANS_IE_SERVICEID);
        String method = iotaMsg.getString(DataTransService.DATATRANS_IE_METHOD);
        String cmdContent =
iotaMsg.getString(DataTransService.DATATRANS_IE_CMDCONTENT);
if (serviceId.equals("switch"))
        {
     //Use the JSON component to parse cmdContent based on the command parameters
defined by the profile.
            //Send command to Switch
        }
        return;
    }
};
mLocalBroadcastManager = LocalBroadcastManager.getInstance(this);
IntentFilter filterReceiveCmd
 = new IntentFilter(DataTransService.TOPIC_COMMAND_RECEIVE);
mLocalBroadcastManager.registerReceiver(mReceiveCmd, filterReceiveCmd);
```

5.2.6 Releasing Data

API Function

This API is used by the gateway to release data of directly connected devices or indirectly connected devices to the IoT platform. In data releasing, users can select topics for the released data, which is not supported in data reporting. In addition, the JSON character strings need to be assembled by the developer and transferred to the API. The SDK is used only for transparent transmission.

API Description

```
mqttDataPub(int cookie, String topic, int qos, byte[] serviceData);
```

Class

DataTransService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	int	The value ranges from 1 to 65535.
pucTopic	Mandatory	String	Specifies the topic of the released data.
uiQos	Mandatory	int	Specifies the value of **MQTT**. Default value: **1**

Parameter	Mandatory or Optional	Type	Description
pbstrServiceData	Mandatory	byte[]	Specifies the packet body of the released data.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the service data is released successfully. The service data is released successfully only after the **DataTransService** broadcast is received.

Example

Call this API to release data.

```
DataTransService. mqttDataPub(1211, "/huawei/v1/devices/336d9bac-9ebf-44e9-95cf-efac5f05da3a/services/Storage", 1, bstrBody);
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class Subscribe implements MyObserver {
    public Subscribe (Observable dataTransService) {
        dataTransService. registerObserver (this);
    }
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
System.out.println ("AgentLiteDataTrans receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
//Receive a data reporting response.
            case IodevService.IODEV_MSG_DATA_REPORT_RSP:
            getDataReportAnswer(arg0);
            break;
//Receive a command passively.
            case IodevService.IODEV_MSG_RECEIVE_CMD:
            getCmdReceive(arg0);
            break;
//MQTT message push
            case IodevService.IODEV_MSG_MQTT_PUB_RSP:
            //logoutResultAction(iotaMsg);
            break;
            case IodevService.IODEV_MSG_MQTT_SUB_RSP:
            //TopicSubcribeResultAction(iotaMsg);
            break;
            case IodevService.IODEV_MSG_MQTT_DATA_RECV_RSP:
            //DataRecvAction(iotaMsg);
            break;
            default:
            break;
```



```
        }
    }
```

5.3 Common Data Structures

5.3.1 IotaDeviceInfo Class

Parameter	Mandatory or Optional	Type	Description
nodeId	Mandatory	String	Uniquely identifies a device managed by the gateway connected to the IoT platform. This parameter is a key parameter determined by the device.
name	Optional	String	Specifies the name of a device.
description	Optional	String	Specifies the description of a device.
manufacturerId	Mandatory	String	Identifies a manufacturer.
manufacturerName	Optional	String	Specifies the name of a manufacturer.
mac	Optional	String	Specifies the MAC address of a device.
location	Optional	String	Specifies the location of a device.
deviceType	Mandatory	String	Specifies the type of a device.
model	Mandatory	String	Specifies the model of a device. ● If a directly connected device is used, the value must be the same as the model defined in the profile. ● If a Z-Wave device is used, the model is in the hexadecimal format of **ProductType** + **ProductId** (padded with zeros if required), for example, **001A-0A12**.
swVersion	Optional	String	Specifies the software version. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String	Specifies the firmware version.
hwVersion	Optional	String	Specifies the hardware version.
protocolType	Mandatory	String	Specifies the protocol type (Z-Wave).
bridgeId	Optional	String	Specifies the bridge through which devices connect to the IoT platform.

Parameter	Mandatory or Optional	Type	Description
status	Optional	String	Specifies the status of a device. The value is **ONLINE** or **OFFLINE**. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline.
statusDetail	Optional	String	Specifies the details about the device status. If **pcStatus** is specified, this parameter is mandatory. Value: ● NONE ● CONFIGURATION_PENDING ● COMMUNICATION_ERROR ● CONFIGURATION_ERROR ● BRIDGE_OFFLINE ● FIRMWARE_UPDATING ● DUTY_CYCLE ● NOT_ACTIVE
mute	Optional	String	Specifies whether a device is screened. ● TRUE ● FALSE

5.3.2 BIND_IE_RESULT IE Parameters

Enumerated Item	Value	Type	Description
BIND_RESULT_SUCCESS	0	NA	Indicates successful binding.
BIND_RESULT_DEV_NOT_BIND	1	NA	Indicates that a device is not bound.
BIND_RESULT_VERIFYCODE_EXPIRED	2	NA	Indicates that the verification code has expired.
BIND_RESULT_FAILED	255	NA	Indicates other failures.

5.3.3 LOGIN_IE_REASON IE Parameters

Enumerated Item	Value	Type	Description
LOGIN_REASON_NULL	0	NA	Indicates that the reason is not specified.
LGN_REASON_CONNCET_ERR	1	NA	Indicates a connection failure.
LOGIN_REASON_SERVER_BUSY	2	NA	Indicates that the server is busy.
LOGIN_REASON_AUTH_FAILED	3	NA	Indicates an authentication failure. Developers need to reconnect the device to the IoT platform.
LOGIN_REASON_NET_UNAVAILABLE	4	NA	Indicates that the network is unavailable.
LOGIN_REASON_DEVICE_NOEXIST	5	NA	Indicates that the device does not exist. Developers need to reconnect the device to the IoT platform.
LOGIN_REASON_DEVICE_RMVED	6	NA	Indicates that the device is deleted. Developers need to reconnect the device to the IoT platform.
LOGIN_REASON_UNKNOWN	7	NA	Indicates an unknown reason.

5.3.4 HUB_IE_RESULT Parameters

Enumerated Item	Value	Type	Description
HUB_RESULT_SUCCESS	0	NA	Indicates that a device is added or deleted.
HUB_RESULT_DEVICE_EXIST	1	NA	Indicates that the device already exists.
HUB_RESULT_DEVICE_NOTEXIST	2	NA	Indicates that the device does not exist.
HUB_RESULT_DEVICE_FAILED	255	NA	Indicates that the execution fails.

6 AgentLite API Reference (C)

6.1 Before You Start

1. Overview

The IoT AgentLite (AgentLite for short) provides standard capabilities for intelligent devices to access the IoT platform in smart home, industrial IoT, and Connected Vehicle fields. These intelligent devices, such as IP cameras (IPCs), lightweight gateways, industrial gateways, and head units, have strong computing capabilities.

2. API Overview

The following table lists the functions provided by the AgentLite, the APIs for implementing these functions, and the relationship between the APIs. It helps developers quickly locate APIs that are used to develop specific services.

Function	API	Description
Access of directly connected devices	IOTA_Init	This API is used to initialize AgentLite resources.
	IOTA_Destroy	Releasing AgentLite Resources
	IOTA_ConfigSetXXX	Binds the configuration.
	IOTA_Bind	Binds devices.
	IOTA_TOPIC_CMD_UNBIND_RECEIVE	Receives device unbinding commands.

Function	API	Description
	IOTA_SetConfig	Configures relevant parameters.
	IOTA_Login	Connects devices to the IoT platform.
	IOTA_Logout	Disconnects devices from the IoT platform.
Management of non-directly connected devices on a gateway	IOTA_HubDeviceAdd	Adds devices.
	IOTA_DeviceStatusUpdate	Updates device status.
	IOTA_HubDeviceRemove	Deletes devices.
Reporting device service data	IOTA_ServiceDataReport	Reports device service data.
Receiving device service commands	IOTA_TOPIC_SERVICE_COMMAND_ RECEIVE/{deviceId}	Receives device service commands.

6.2 API Description

This section describes the external APIs provided by the AgentLite that can be used for broadcast mechanism, access of directly connected devices, management of non-directly connected devices on a gateway, reporting device service data, receiving device commands, and describing JSON component.

6.2.1 Broadcast Mechanism

The AgentLite provides a broadcast mechanism for third-party developers to receive messages reported by the AgentLite.

Subscribing to a Broadcast

```
HW_BroadCastReg(HW_CHAR *pcTopic ,PFN_HW_BROADCAST_RECV pfnReceiver);
```

The broadcast processing function prototype is as follows:

```
(*PFN_HW_BROADCAST_RECV)(HW_UINT uiCookie, HW_MSG *pstMsg);
```

uiCookie in the broadcast is the same as **uiCookie** provided over the API. It is used to match the request and response of a service. If **uiCookie** is not provided over the API, **uiCookie** in the broadcast is invalid.

Unsubscribing from a Broadcast

```
HW_BroadCastUnreg(HW_CHAR *pcTopic, PFN_HW_BROADCAST_RECV pfnReceiver);
```

Obtaining Data from the pstMsg

Obtaining a character string:


```
HW_MsgGetStr(HW_MSG pstMsg, HW_UINT uiTag);
```

Obtaining an unsigned integer:

```
HW_MsgGetUint(HW_MSG pstMsg, HW_UINT uiTag, HW_UINT uiDefault);
```

Obtaining a byte array:

```
HW_MsgGetByteArray(HW_MSG pstMsg, HW_UINT uiTag);
```

6.2.2 Access of Directly Connected Devices

6.2.2.1 Initializing AgentLite Resources

API Function

This API is used to initialize AgentLite resources.

API Description

```
HW_INT IOTA_Init(const HW_CHAR *pcWorkPath, const HW_CHAR *pcLogPath);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
pcWorkPath	Mandatory	String	Specifies the AgentLite working path for storing AgentLite configuration files and generated temporary files. The working path must be valid and end with \0.
pcLogPath	Optional	String	Specifies the log path. (If this parameter is left unspecified, logs are written to the working path.) The log path must be end with \0.

Return Value

For details, see **4.2 Function Return Values**.

Example

```
//Developers call this API to initialize AgentLite resources.
IOTA_Init("/usr/data", HW_NULL);
```

6.2.2.2 Releasing AgentLite Resources

API Function

This API is used to release all dynamic resources that are applied for, such as the memory and thread.

API Description

```
IOTA_VOID IOTA_Destroy();
```

Return Value

For details, see **4.2 Function Return Values**.

Example

```
//Developers call this API to release AgentLite resources.
IOTA_Destroy();
```

6.2.2.3 Binding the Configuration

API Function

This API is used to configure AgentLite parameters.

API Description

```
HW_INT IOTA_ConfigSetStr(HW_INT iItem, HW_CHAR *pValue)
HW_INT IOTA_ConfigSetUint(HW_INT iItem, HW_UINT uiValue)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	HW_UINT	Specifies the configuration item bound to the device. ● Platform IP address: EN_IOTA_CFG_IOCM_ADDR ● Platform port number: EN_IOTA_CFG_IOCM_PORT
value	Mandatory	HW_CHAR *	Specifies the value of the configuration item. ● Platform IP address: IP address of the AgentLite for interworking with the platform. ● Platform port number: 8943

Return Value

For details, see **4.2 Function Return Values**.

Example

```
//Developers call this API to configure parameters.
IOTA_ConfigSetStr (EN_IOTA_CONFIG_IOCM_ADDR, "10.0.0.1");
IOTA_ConfigSetUint(EN_IOTA_CFG_IOCM_PORT, 8943);
```

6.2.2.4 Binding a Device

API Function

A device must be bound to the IoT platform before accessing the IoT platform for the first time. The upper-layer application calls this API to transfer the device serial number, MAC address, or other device information to bind a device to the IoT platform.

Before binding a device, developers must call **Configuring Service Parameters** carrying the mandatory parameters **EN_IOTA_CFG_IOCM_ADDR** and **EN_IOTA_CFG_IOCM_PORT** to set the IP address and port number of the server to be bound. (For the IoCM server, the default port 8943 is configured on the AgentLite.)

NOTE

Before a directly connected device accesses the IoT platform for the first time, developers must register the device on the IoT platform and then initiate a binding request on the device. If the device is not registered on the IoT platform, the binding fails. The AgentLite waits for a while and tries again.

API Description

```
HW_INT IOTA_Bind(const HW_CHAR *pcVerifyCode, const ST_IOTA_DEVICE_INFO *pstInfo);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
pcVerifyCode	Mandatory	String	Specifies a device binding verification code. The value must end with \0. If the device is registered on the SP portal, set **pcVerifyCode** to the preSecret used during device registration.
pstInfo	Mandatory	**ST_IOTA_DEVICE_INFO** structure	Specifies the device information. The value must end with \0.

Return Value

For details, see **4.2 Function Return Values**.

NOTE

- Return values only show API calling results. For example, return value **0** indicates that the API is called successfully but does not indicate that the connection is successful. The connection is successful only after the **IOTA_TOPIC_BIND_RSP** broadcast is received.
- If the binding fails, the API automatically binds the device after 30 seconds. If the retry fails for consecutive five times (the total number of attempts is six), the API returns a message indicating that the binding fails and stops the binding. To enable the API to retry again, the user needs to restart the device.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_BIND_RSP	HW_MSG object	**EN_IOTA_BIND_IE_TYPE**	Indicates the binding result. For details, see **EN_IOTA_BIND_IE_TYPE enumeration description**.

Example

Bind the device by calling IOTA_Bind().

```
//Developers call this API to bind a device.
ST_HW_DEVICE_INFO stDeviceInfo
stDeviceInfo.pcNodeId = "SN Number";
stDeviceInfo.pcManufacturerId = "Huawei";
stDeviceInfo.pcDeviceType = "Gateway";
stDeviceInfo.pcModel = "HW_GW101";
stDeviceInfo.pcProtocolType = "HuaweiM2M";

IOTA_Bind("SN Number", &stDeviceInfo);
```

After a device is bound, the AgentLite returns the parameters shown in the following output. Developers must store and configure these parameters on the device before connecting the device to the IoT platform.

```
//Register the broadcast receive processing function.
HW_BroadCastReg("IOTA_TOPIC_BIND_RSP", Device_RegResultHandler);
//Developers call this API to register the binding result returned by the
function.
HW_iNT Device_RegResultHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    HW_cHAR *pcDeviceId;
    HW_cHAR *pcDeviceSecret;
    HW_cHAR *pcAppId;
    HW_cHAR *pcIoCMServerAddr;
    HW_UiNT uiIoCMServerPort;
    HW_cHAR *pcMqttServerAddr;
    HW_UiNT uiMqttServerPort;
    If (HW_SUCCESS != HW_MsgGetUint(pstMsg, EN_IOTA_BIND_IE_RESULT, 0))
    {
        Return 0;
    }
    pcDeviceId = HW_MsgGetStr(pstMsg, EN_IOTA_BIND_IE_DEVICEID);
    pcDeviceSecret = HW_MsgGetStr(pstMsg, EN_IOTA_BIND_IE_DEVICESECRET);
    pcAppId = HW_MsgGetStr(pstMsg, EN_IOTA_BIND_IE_APPID);
```



```
    pcIoCMServerAddr = HW_MsgGetStr(pstMsg, EN_IOTA_ BIND_IE_IOCM_ADDR );
    uiIoCMServerPort = HW_MsgGetUint(pstMsg, EN_IOTA_BIND_IE_IOCM_PORT, 0);
    pcMqttServerAddr = HW_MsgGetStr(pstMsg, EN_IOTA_ BIND_IE_IOCM_ADDR );
    uiMqttServerPort = HW_MsgGetUint(pstMsg, EN_IOTA_BIND_IE_IOCM_PORT, 0);
Config_save("DeviceId", pcDeviceId);
Config_save("DeviceSecret", pcDeviceSecret);
Config_save("AppId", pcAppId);
Config_save("IoCMAddr", pcIoCMServerAddr);
Config_save("IoCMPort", pcIoCMServerPort);
Config_save("MqttAddr", pcMqttServerAddr);
Config_save("MqttPort", pcMqttServerPort);
    return 0;
}
```

6.2.2.5 Unbinding a Device

API Function

This API is used to register the broadcast for receiving the command used to unbind a directly connected device from the IoT platform. After this broadcast is received, developers must delete the configuration information about the directly connected device and release all resources. When the device restarts, developers must bind the device to the IoT platform again.

API Description

```
IOTA_TOPIC_CMD_UNBIND_RECEIVE;
```

Parameter Description

None

Output

None

Example

```
//Developers call this API to register the function to unbind a directly
connected device.
HW_iNT Gateway_UnbindRecvtHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
// Delete Config file, free resources.
    return 0;
}
//Register the function HW_BroadCastReg ("IOTA_TOPIC_CMD_UNBIND_RECEIVE",
Gateway_UnbindRecvtHandler) during initialization;
```

6.2.2.6 Configuring Parameters

6.2.2.6.1 Configuring Service Parameters

API Function

This API is used to set the parameters required for login.

API Description

```
HW_INT IOTA_ConfigSetStr(HW_INT iItem, HW_CHAR *pValue)
HW_INT IOTA_ConfigSetUint(HW_INT iItem, HW_UINT uiValue)
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	int	Specifies the configuration item for logging in to the device. ● Device ID: EN_IOTA_CFG_DEVICEID. ● App ID: EN_IOTA_CFG_APPID. ● Secret: EN_IOTA_CFG_DEVICESECRET. ● HTTP address: LEN_IOTA_CFG_IOCM_ADDR. ● HTTP port: EN_IOTA_CFG_IOCM_PORT. ● MQTT address: EN_IOTA_CFG_MQTT_ADDR. ● MQTT port: EN_IOTA_CFG_MQTT_PORT.
value	Mandatory	String	Specifies the value of the configuration item. ● Device ID: obtain the value from the callback that is bound. ● App ID: obtain the value from the callback that is bound. ● Secret: obtain the value from the callback that is bound. ● HTTP address: southbound access address for the AgentLite to connect to the IoT platform. ● HTTP port: 8943. ● MQTT address: southbound access address for the AgentLite to connect to the IoT platform. ● MQTT port: 8883.

Return Value

For details, see **4.2 Function Return Values**.

Example

```
//Developers call this API to configure parameters.
IOTA_ConfigSetStr (EN_IOTA_CONFIG_IOCM_ADDR, "10.0.0.1");
IOTA_ConfigSetUint(EN_IOTA_CFG_IOCM_PORT, 8943);
```

6.2.2.6.2 (Optional) Configuring Encryption Algorithm Type Parameters

API Function

This API is used to configure the encryption algorithm type for sensitive information before login. The service can select an appropriate encryption algorithm type based on the security level.

API Description

```
HW_UINT HW_SetAlgType(HW_UINT uiAlgType);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiAlgType	Mandatory	Int	Specifies the encryption algorithm type. The value ranges from 0 to 2. ● **0**: AES 256 CBC. ● **1**: AES 128 GCM ● **2**: AES 256 GCM.

Return Value

For details, see **4.2 Function Return Values**.

Example

```
//Developers call this API to configure parameters.
HW_SetAlgType(2);
```

6.2.2.7 Device Login

API Function

This API is used to connect a device to the IoT platform after the device is bound to the IoT platform for the first time or the device restarts.

API Description

```
HW_INT IOTA_Login();
```

Parameter Description

None

Return Value

For details, see **4.2 Function Return Values**.

NOTE

Return values only show API calling results. For example, return value **0** indicates that the API is called successfully but does not indicate that the connection is successful. The connection is successful only after the **IOTA_TOPIC_CONNECTED_NTY** broadcast is received. Before calling the **HW_INT IOTA_Login()** API, call the **IOTA_SetConfig** API to configure relevant parameters.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_CONNECTED_NTY	HW_MSG object	None	Specifies that the connection or reconnection is successful.
IOTA_TOPIC_DISCONNECT_NTY	HW_MSG object	EN_ULGN_IE_ERR_REASON	Specifies that the connection fails or the device is disconnected from the IoT platform.

Example

```
Config_Get("DeviceId", pcDeviceId);
Config_Get("DeviceSecret", pcDeviceSecret);
Config_Get("AppId", pcAppId);
Config_Get("HAAddr", pcHAServerAddr);
Config_Get("LVSAddr", pcLVSServerAddr);

IOTA_SetConfig(EN_IOTA_CFG_DEVICEID, pcDeviceId);
IOTA_SetConfig(EN_IOTA_CFG_DEVICESECRET, pcDeviceSecret);
IOTA_SetConfig(EN_IOTA_CFG_APPID, pcAppId);
IOTA_SetConfig(EN_IOTA_CFG_HA_ADDR, pcHAServerAddr);
IOTA_SetConfig(EN_IOTA_CFG_LVS_ADDR, pcLVSServerAddr);

IOTA_Login();
```

The device waits for a connection status broadcast from the AgentLite.

Developers need to register the functions for processing connection status broadcasts in advance. It is recommended that:

1. Gateways report the status of indirectly connected devices and all the cached data when the connection is successful.
2. Gateways record the disconnected status of devices and cache the reported data when the connection fails.

```
//Developers call this API to register the function for subsequent processing
after successful connection.
HW_iNT Device_ConnectedHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    //update device states
    //send buffer data
    return 0;
```



```
}
//Developers call this API to register the function for subsequent processing
after the connection fails.
HW_iNT Device_DisconnectHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    //stop reporting data
    return 0;
}
//Developers call this API to bind the broadcast processing function.
HW_BroadCastReg("IOTA_TOPIC_CONNECTED_NTY", Device_ConnectedHandler);
HW_BroadCastReg("IOTA_TOPIC_DISCONNECT_NTY", Device_DisconnectHandler);
```

After the login is successful, the device is connected to the IoT platform.

If the device is disconnected from the IoT platform due to the network or server fault, the AgentLite will automatically attempt to reconnect the device to the IoT platform and report the real-time status to the third-party application by using the broadcasts **IOTA_TOPIC_CONNECTED_NTY** and **IOTA_TOPIC_DISCONNECT_NTY**.

6.2.2.8 Device Logout

API Function

This API is used to disconnect a device from the IoT platform.

API Description

```
HW_INT IOTA_Logout();
```

Parameter Description

None

Return Value

For details, see **4.2 Function Return Values**.

NOTE

Return values only show API calling results. For example, return value **0** indicates that the API is called.

Example

```
//Developers call this API to disconnect the device.
IOTA_Logout();
```

6.2.3 Management of Non-Directly Connected Devices on a Gateway

If the device on which the AgentLite is integrated is a gateway, it needs to manage all non-directly connected devices (sensors), including access and deletion of these devices. In addition, the gateway needs to record the mapping between logical and physical IDs of these devices.

6.2.3.1 Adding a Device

API Function

This API is used to add a device that accesses the IoT platform through the gateway. After the device is connected to the IoT platform, the IoT platform will assign a unique logical ID to the device.

API Description

```
HW_INT IOTA_HubDeviceAdd(HW_UINT uiCookie, const ST_IOTA_DEVICE_INFO
*pstDeviceInfo);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	HW_UINT	The value ranges from 1 to 65535.
pstDeviceInfo	Mandatory	**ST_IOTA_DEVICE_INFO** structure	Specifies the device information. The value must end with \0.

Return Value

For details, see **4.2 Function Return Values**.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_HUB_ADDDEV_RSP	HW_MSG object	**EN_IOTA_HUB_IE_TYPE**	Specifies the device addition result. If the addition is successful, the device ID is returned.

Example

```
//Developers call this API to add a device.
ST_IOTA_DEVICE_INFO stDeviceInfo
stDeviceInfo.pcNodeId = "SN Number";
stDeviceInfo.pcManufacturerId = "Huawei";
stDeviceInfo.pcDeviceType = "Camera";
stDeviceInfo.pcModel = "HW_CAM101";
stDeviceInfo.pcProtocolType = "ONVIF";

IOTA_HubDeviceAdd(29011, &stDeviceInfo);
```

The device waits for the result.

```
//Developers call this API to register the function for subsequent processing
after the device is added.
HW_iNT Device_AddResultHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    uiResult = HW_MsgGetUint(pstMsg, EN_IOTA_HUB_IE_RESULT);
    if (EN_IOTA_HUB_RESULT_SUCCESS != uiResult)
    {
        // retry with uiCookie
        return 0;
    }
    return 0;
}
//Bind the broadcast reception processing function HW_BroadCastReg
("IOTA_TOPIC_HUB_ADDDEV_RSP", Device_AddResultHandler).
```

6.2.3.2 Updating the Device Status

API Function

This API is used to update status information about devices, including directly connected devices and indirectly connected devices managed by the IoT platform. The device status can be updated through this API when the device is offline or online.

The status of directly connected devices is updated based on the device login status. If a directly connected device is disconnected from the IoT platform, its status becomes offline. If a directly connected device is connected or reconnected to the IoT platform, its status becomes online and does not need to be updated through this API. Therefore, it is recommended that developers call this API to update the status of indirectly connected devices.

API Description

```
HW_INT IOTA_DeviceStatusUpdate(HW_UINT uiCookie, const HW_CHAR *pcDeviceId, const
HW_CHAR *pcStatus, const HW_CHAR *pcStatusDetail);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	HW_UINT	The value ranges from 1 to 65535.
pcDeviceId	Mandatory	HW_CHAR	Identifies a device. The value must end with \0.

Parameter	Mandatory or Optional	Type	Description
pcStatus	Mandatory	HW_CHAR	Specifies the device status. The value must end with \0. The value is **ONLINE** or **OFFLINE**. ● **ONLINE**: indicates that the device is in the online state. ● **OFFLINE**: indicates that the device is in the offline state.
pcStatusDetail	Mandatory	HW_CHAR	Specifies the detailed device status information. The value must end with \0. Value: ● **None** ● **CONFIGURATION_PENDING** ● **COMMUNICATION_ERROR** ● **CONFIGURATION_ERROR** ● **BRIDGE_OFFLINE** ● **FIRMWARE_UPDATING** ● **DUTY_CYCLE** ● **NOT_ACTIVE**

Return Value

For details, see **4.2 Function Return Values**.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_DEVUPDATE_RSP/{deviceId}	HW_MSG object	None	Specifies the device status update result.

Example

```
HW_CHAR *pcDeviceId = stDevice.pcDeviceId;

IOTA_DeviceStatusUpdate(0, pcDeviceId, "ONLINE", "NONE");
```

The device waits for the command execution result.

```
//Developers call this API to register the function for subsequent processing
after the device is updated.
HW_iNT Device_StatusUpdateHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    HW_cHAR pcCmdContentl;
    pcCmdContent = HW_MsgGetStr(pstMsg, EN_IOTA_DEVUPDATE_IE_RESULT);
    pcCmdContent = HW_MsgGetStr(pstMsg, EN_IOTA_DEVUPDATE_IE_DEVICEID);
    return 0;
}
//Bind the broadcast reception processing function HW_BroadCastReg
("IOTA_TOPIC_DEVUPDATE_RSP", Device_StatusUpdateHandler).
```

6.2.3.3 Deleting a Device

API Function

This API is used to delete indirectly connected devices from the IoT platform when a new device needs to be removed from the gateway.

API Description

```
HW_INT IOTA_HubDeviceRemove(HW_UINT uiCookie, const HW_CHAR *pcDeviceId);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	HW_UINT	The value ranges from 1 to 65535.
pcDeviceId	Mandatory	String	Identifies a device. The value must end with \0.

Return Value

For details, see **4.2 Function Return Values**.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_HUB_RMVDEV_RSP	HW_MSG object	EN_IOTA_HUB_IE_TYPE	Specifies the device deletion result.

Example

```
//Developers call this API to delete a device.
HW_cHAR *pcDeviceId = stDevice.pcDeviceId;
IOTA_HubDeviceRemove(HW_NULL, pcDeviceId);
The device then waits for the deletion result.
HW_iNT Device_RemoveResultHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    uiResult = HW_MsgGetUint (pstMsg, EN_IOTA_HUB_IE_RESULT);
    if (EN_IOTA_HUB_RESULT_SUCCESS != uiResult)
    {
        // retry with uiCookie
        return 0;
    }
    return 0;
}
HW_BroadCastReg("IOTA_TOPIC_HUB_RMVDEV_RSP", Device_RemovResultHandler);
```

6.2.4 Reporting Device Service Data

API Function

This API is used by the gateway to report data of directly connected devices or indirectly connected devices to the IoT platform.

API Description

```
HW_INT IOTA_ServiceDataReport(HW_UINT uiCookie, const HW_CHAR *pcRequstId,
const HW_CHAR *pcDeviceId, const HW_CHAR *pcServiceId, const HW_CHAR
*pcServiceProperties);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	unsign int	The value ranges from 1 to 65535.
pcRequstId	Mandatory	String	Identifies a request. The request ID is used to match the service command delivered by the IoT platform. This parameter is mandatory if the reported data is the response to the command request initiated by the IoT platform. The value must end with \0.

Parameter	Mandatory or Optional	Type	Description
pcDeviceId	Mandatory	String	Identifies a device. The value must end with \0.
pcServiceId	Mandatory	String	Identifies a service. The value must end with \0.
pcServiceProperties	Mandatory	String	Specifies the service attribute. The value must end with \0.

Return Value

For details, see **4.2 Function Return Values**.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_DATATRANS_REPORT_RSP/{deviceId}	HW_MSG object	EN_IOTA_DATATRANS_IE_RESULT	Specifies the data reporting result. The value is **0** or **1**. ● **0**: indicates that the data is reported. ● **1**: indicates that the data reporting fails.

Example

A user uses JSON components to assemble service attributes (pcServiceProperties) based on the format of profiles.

```
HW_UINT *uiLen;

IOTA_ServiceDataReport(1211, NULL, "xxxx_xxxx_xxxx_xxxx" , "DoorWindow",
"{\"status\":\"OPEN\"}");
```

The device waits for the data reporting result.

```
//Developers call this API to register the function for subsequent processing
after the device service data is reported.
HW_iNT Device_DataReportResultHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    uiResult = HW_MsgGetUint(pstMsg, EN_IOTA_DATATRANS_IE_RESULT);
    if (HW_SUCCESS != uiResult)
    {
        // retry with uiCookie
        return 0;
    }
```

```
    return 0;
}
//After the device is added, register a broadcast to receive the service data
reporting result: HW_BroadCastReg ("IOTA_TOPIC_DATATRANS_REPORT_RSP/
XXXX_XXXX_XXXX_XXXX", Device_AddResultHandler).
```

6.2.5 Receiving Device Commands

API Function

This API is used to register the broadcast for receiving control commands from the IoT platform.

API Description

```
IOTA_TOPIC_SERVICE_COMMAND_RECEIVE/{deviceId};
```

Parameter Description

For details, see **IEs in the EN_IOTA_DATATRANS_IE_TYPE Messages**.

Output

None

Example

```
//Developers call this API to register the function for subsequent processing
after device commands are received.
HW_iNT Switch_CommandRecvtHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    HW_cHAR *pcMethod, *pcServiceId, *pcCmdContent, *pcDeviceId;
    pcDeviceId = HW_MsgGetStr(pstMsg, EN_IOTA_DATATRANS_IE_DEVICEID);
    pcServiceId = HW_MsgGetStr(pstMsg, EN_IOTA_DATATRANS_IE_SERVICEID);
    pcMethod = HW_MsgGetStr(pstMsg, EN_IOTA_DATATRANS_IE_METHOD);
    pcCmdContent = HW_MsgGetStr(pstMsg, EN_IOTA_DATATRANS_IE_CMDCONTENT);
if (strcmp(pcServiceId, "switch"))
    {
//Developers call this API to use the JSON components to parse pcCmdContent based
on command parameters in the profile.
        //Send command to Switch
    }
    return 0;
}
//Developers call this API to register the broadcast for receiving device
commands immediately after the device is added.
HW_BroadCastReg("IOTA_TOPIC_SERVICE_CMD_RECEIVE/XXXX_XXXX_XXXX_XXXX",
Device_AddResultHandler);
```

After a device is added, developers must register the broadcast for receiving commands for the device. The broadcast is in format of **IOTA_TOPIC_SERVICE_CMD_RECEIVE/Device ID**. When the AgentLite receives commands delivered by the IoT platform to the device, it broadcasts these commands to the broadcast processing function registered by the device. If the commands do not need to be dispatched by device ID, the broadcast can be in format of **IOTA_TOPIC_SERVICE_CMD_RECEIVE**.

6.2.6 Releasing Data

API Function

This API is used by the gateway to release data of directly connected devices or indirectly connected devices to the IoT platform. For data releasing scenarios, the released data can be defined as a topic. In addition, the package body needs to be assembled to the input parameters. The data is transparently transmitted internally instead of being assembled.

API Description

```
HW_INT IOTA_MqttDataPub (HW_UINT uiCookie, const HW_UCHAR *pucTopic, HW_UINT
uiQos, const HW_BYTES *pbstrServiceData);
```

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	unsign int	The value ranges from 1 to 65535.
pucTopic	Mandatory	String	Specifies the topic for releasing data. The value must end with \0.
uiQos	Mandatory	unsign int	Specifies the value of **MQTT**. Default value: **1**
pbstrServiceData	Mandatory	HW_BYTES	Specifies the package for releasing data.

Return Value

For details, see **4.2 Function Return Values**.

Output

Broadcast Name	Broadcast Parameter	Member	Description
IOTA_TOPIC_MQTT_DATA_PUB_RSP	HW_MSG object	EN_IOTA_DATATRANS_IE_RESULT	Specifies the data release result. ● **0**: indicating that the data is released. ● **1**: indicates that the data releasing fails.

Example

```
HW_BYTES bstrBody;
...
IOTA_MqttDataPub (1211, "/huawei/v1/devices/336d9bac-9ebf-44e9-95cf-efac5f05da3a/
services/Storage", 1, bstrBody);
```

The device waits for the data reporting result.

```
//Developers call this API to register the function for subsequent processing
after the device service data is reported.
HW_iNT Device_DataPubResultHandler(HW_UiNT uiCookie, HW_MSG pstMsg)
{
    uiResult = HW_MsgGetUint(pstMsg, EN_IOTA_DATATRANS_IE_RESULT);
    if (HW_SUCCESS != uiResult)
    {
        // retry with uiCookie
        return 0;
    }
    return 0;
}
//After the device is added, register a broadcast to receive the service data
reporting result: HW_BroadCastReg("IOTA_TOPIC_MQTT_DATA_PUB_RSP",
Device_AddResultHandler);
```

6.2.7 JSON Component Description

The JSON component is provided by the AgentLite to developers for encoding and decoding data in JSON format. It is used to assemble reported data and deliver command parsing results.

1. JSON Encoding

The JSON encoding process is as follows:

Create a JSON coding object.

```
HW_JSONOBJ HW_JsonObjCreate()
```

Obtain the root node of the JSON object.

```
HW_JSON HW_JsonGetJson(HW_JSONOBJ hjson)
```

Add key-value pairs to the JSON object.

Add the key-value pair in which the pcVal is a character string.


```
HW_INT HW_JsonAddStr(HW_JSON pstJson, HW_CHAR *pcKey, HW_CHAR *pcVal)
```

Add the key-value pair in which the uiVal is an integer.

```
HW_INT HW_JsonAddUint(HW_JSON pstJson, HW_CHAR *pcKey, HW_UINT uiVal)
```

Add the key-value pair in which the bVal is a Boolean value.

```
HW_INT HW_JsonAddBool(HW_JSON pstJson, HW_CHAR *pcKey, HW_BOOL bVal)
```

Add the key-value pair whose value is a JSON object to obtain a sub-JSON object.

```
HW_JSON HW_JsonAddJson(HW_JSON pstJson, HW_CHAR *pcKey)
```

Add the key-value pair whose value is a JSON array to obtain a sub-JSON array object.

```
HW_JSON_ARRAY HW_JsonAddArray(HW_JSON pstJson, HW_CHAR *pcKey)
```

Add key-value pairs to the JSON array.

Add the key-value pair in which the pcVal is a character string.

```
HW_INT HW_JsonArrayAddStr(HW_JSON_ARRAY *pstArray, HW_CHAR *pcKey, HW_CHAR *pcVal)
```

Add the key-value pair in which the uiVal is an integer.

```
HW_INT HW_JsonArrayAddUint(HW_JSON_ARRAY *pstArray, HW_CHAR *pcKey, HW_UINT uiVal)
```

Add the key-value pair in which the bVal is a Boolean value.

```
HW_INT HW_JsonArrayAddBool(HW_JSON_ARRAY *pstArray, HW_CHAR *pcKey, HW_BOOL bVal)
```

Add the key-value pair in which the pucValue is a JSON object to obtain a sub-JSON object.

```
HW_JSON HW_JsonArrayAddJson(HW_JSON_ARRAY pstArray)
```

Add the key-value pair in which the pucValue is a JSON array to obtain a sub-JSON array object.

```
HW_JSON_ARRAY *HW_JsonArrayAddArray(HW_JSON_ARRAY *pstArray)
*HW_JsonArrayAddArray(HW_JSON_ARRAY *pstArray)
```

Obtain JSON character strings.

```
HW_CHAR *HW_JsonEncodeStr(HW_JSONOBJ hJson);
```

Delete the JSON object.

```
HW_VOID HW_JsonObjDelete(HW_JSONOBJ *phJson);
```

Encoding Example:

A JSON message to be parsed is as follows:

```
{
    "temperature":22,
    "otherInfo":{
        "batteryLevel":"low"
    }
}
/*Define variables.*/
HW_JSONOBJ jsonObj;
HW_JSON rootjson;
HW_JSON json;
HW_CHAR *pcJsonStr;

/*Create a JSON encoding object.*/
hJsonObj = HW_JsonObjCreate();

/*Obtain the root node of the JSON object. */
```



```
rootjson = HW_JsonGetJson(hJsonObj);

/*Add a key-value pair to the root node.*/
HW_JsonAddUint(rootjson, "temperature", 22);

/*Obtain the sub-JSON object from the root node.*/
json = HW_JsonAddJson(rootjson, "otherInfo");

/*Add a key-value pair to the sub-JSON object.*/
HW_JsonAddStr(json, " batteryLevel", "low");


/*Obtain JSON character strings. */
pcJsonStr = HW_JsonEncodeStr(hjsonObj);

/*Delete the JSON encoding object that was previously created, to release
resources.*/
HW_JsonObjDelete(&hJsonObj);
```

2. JSON Decoding

The JSON decoding process is as follows:

Create a JSON parsing object.

```
HW_JSONOBJ HW_JsonDecodeCreate(HW_CHAR *pucStr, HW_BOOL bStrCpy)
```

Obtain JSON data from the JSON parsing object.

```
HW_JSON HW_JsonGetJson(HW_JSONOBJ hJson)
```

Obtain the character string mapping to the pucKey from the JSON data.

```
HW_CHAR *HW_JsonGetStr(HW_JSON pstJson, HW_CHAR *pucKey)
```

Obtain the unsigned integer mapping to the pucKey from the JSON data.

```
HW_UINT HW_JsonGetUint(HW_JSON pstJson, HW_CHAR *pucKey, HW_UINT uiDft)
```

Obtain the Boolean value mapping to the pucKey from the JSON data.

```
HW_BOOL HW_JsonGetBool(HW_JSON pstJson, HW_CHAR *pucKey, HW_BOOL bDft)
```

Obtain the array mapping to the pucKey from the JSON data.

```
HW_UJSON_ARRAY HW_JsonGetArray(HW_JSON pstJson, HW_CHAR *pucKey)
```

Obtain the length of the JSON array.

```
HW_UINT HW_JsonArrayGetCount(HW_UJSON_ARRAY pstArray)
```

Obtain JSON data mapping to the uiIndex from the JSON array.

```
HW_JSON HW_JsonArrayGetJson(HW_UJSON_ARRAY pstArray, HW_UINT uiIndex)
```

Obtain the unsigned integer mapping to the uiIndex from the JSON array.

```
HW_UINT HW_JsonArrayGetUint(HW_UJSON_ARRAY pstArray, HW_UINT uiIndex, HW_UINT
uiDft)
```

Obtain the Boolean value mapping to the uiIndex from the JSON array.

```
HW_UINT HW_JsonArrayGetBool(HW_UJSON_ARRAY pstArray, HW_UINT uiIndex, HW_BOOL
bDft)
```

Obtain the character string mapping to the uiIndex from the JSON array.

```
HW_CHAR *HW_JsonArrayGetStr(HW_UJSON_ARRAY pstArray, HW_UINT uiIndex)
```


Obtain the sub-array mapping to the uiIndex from the JSON array.

```
HW_UJSON_ARRAY HW_JsonArrayGetArray(HW_UJSON_ARRAY pstArray, HW_UINT uiIndex)
```

Delete the JSON parsing object that was previously created.

```
HW_VOID HW_JsonObjDelete(HW_JSONOBJ *phJson)
```

Parsing Example:

A JSON message to be parsed is as follows:

```
{
    "action":"notify",
    "type":"userstate",
    "userstateinfo":[
        {
            "num":"11111 ",
            "state":"idle"
        },
        {
            "num":"11111",
            "state":"ringing"
        }
    ]
}
/*Define variables.*/
HW_JSONOBJ jsonObj;
HW_JSON json;
HW_UJSON_ARRAY jsonArray;
HW_cHAR *action;
HW_cHAR *type;
HW_UiNT count;
HW_UiNT index;
/*Create a JSON parsing object.*/
jsonObj = HW_JsonDecodeCreate(jsonStr, HW_TRUE);
/*Obtain JSON data from the JSON parsing object.*/
json = HW_JsonGetJson(jsonObj);
/*Obtain the character string mapping to the action from the JSON data.*/
action = HW_JsonGetStr(json, "action");
/*Obtain the character string mapping to the type from the JSON data.*/
type = HW_JsonGetStr(json, "type");
/*Obtain the JSON array mapping to the userstateinfo from the JSON data.*/
jsonArray = HW_JsonGetArray(json, "userstateinfo");
/*Obtain the length of the jsonArray.*/
count = HW_JsonArrayGetCount(jsonArray);
for (index = 0; index < count; index++)
{
/*Obtain JSON data mapping to the index from the jsonArray.*/
    HW_JSON jsonItem = HW_JsonArrayGetJson(jsonArray, index);
/*Obtain the character string mapping to the num from the jsonItem.*/
    HW_cHAR *num = HW_JsonGetStr(jsonItem, "num");
/*Obtain the character string mapping to the state from the jsonItem.*/
    HW_cHAR *state = HW_JsonGetStr(jsonItem, "state");
    ......
}
/*Delete the JSON parsing object that was previously created, to release
resources.*/
HW_JsonObjDelete(jsonObj);
```

6.3 Common Data Structures

6.3.1 ST_IOTA_DEVICE_INFO Structure

Parameter	Mandatory or Optional	Type	Description
pcNodeId	Mandatory	String	Uniquely identifies a device managed by the gateway connected to the IoT platform. This parameter is a key parameter.
pcName	Optional	String	Specifies the name of a device.
pcDescription	Optional	String	Specifies the device description.
pcManufacturerId	Mandatory	String	Identifies a manufacturer.
pcManufacturerName	Optional	String	Specifies the name of a manufacturer.
pcMac	Optional	String	Specifies the MAC address of a device.
pcLocation	Optional	String	Specifies the location of a device.
pcDeviceType	Mandatory	String	Specifies the type of a device.
pcModel	Mandatory	String	Specifies the model of a device. ● If a directly connected device is used, the value must be the same as the model defined in the profile. ● If Z-Wave is used, the model is a hexadecimal number in the format of **ProductType** + **ProductId** (padded with zeros if required), for example, **001A-0A12**.
pcSwVersion	Optional	String	Specifies the software version. When Z-Wave is used, the software version is in the format of major version.minor version, for example, **1.1**.
pcFwVersion	Optional	String	Specifies the firmware version.
pcHwVersion	Optional	String	Specifies the hardware version.
pcProtocolType	Mandatory	String	Specifies the protocol type (Z-Wave).
pcBridgeId	Optional	String	Specifies the bridge through which devices connect to the IoT platform.

Parameter	Mandatory or Optional	Type	Description
pcStatus	Optional	String	Specifies the status of a device. The value is **ONLINE** or **OFFLINE**. ● **ONLINE**: indicates that the device is online. ● **OFFLINE**: indicates that the device is offline.
statusDetail	Optional	String	Specifies the details about the device status. If **pcStatus** is specified, this parameter is mandatory. Value: ● None ● CONFIGURATION_PENDING ● COMMUNICATION_ERROR ● CONFIGURATION_ERROR ● BRIDGE_OFFLINE ● FIRMWARE_UPDATING ● DUTY_CYCLE ● NOT_ACTIVE
pcMute	Optional	String	Specifies whether a device is screened. ● **TRUE**: indicates that the device is screened. ● **FALSE**: indicates that the device is not screened.

6.3.2 IEs in EN_IOTA_BIND_IE_TYPE Messages

Enumerated Item	Value	Type	Description
EN_IOTA_BIND_IE_RESULT	0	**EN_IOTA_BIND_RESULT_TYPE**	Specifies the binding result.
EN_IOTA_BIND_IE_DEVICEID	1	String	Identifies the logical device assigned by the IoT platform.
EN_IOTA_BIND_IE_DEVICESECRET	2	String	Specifies the authentication key for a device to access the IoT platform.

Enumerated Item	Value	Type	Description
EN_IOTA_BIND_IE_APPID	3	String	Identifies the application of a developer.
EN_IOTA_BIND_IE_IOCM_ADDR	4	String	Specifies the IP address of the IoCM server.
EN_IOTA_BIND_IE_IOCM_PORT	5	unsigned int	Specifies the port number of the IoCM server.
EN_IOTA_BIND_IE_MQTT_ADDR	6	String	Specifies the IP address of the MQTT server.
EN_IOTA_BIND_IE_MQTT_PORT	7	unsigned int	Specifies the port number of the MQTT server.

6.3.3 Enumerated Values of the EN_IOTA_BIND_RESULT_TYPE Parameter

Enumerated Item	Value	Type	Description
EN_IOTA_BIND_RESULT_SUCCESS	0	NA	Specifies successful binding.
EN_IOTA_BIND_RESULT_DEV_NOT_BIND	1	NA	Indicates that a device is not bound.
EN_IOTA_BIND_RESULT_VERIFYCODE_EXPIRED	2	NA	Indicates that the verification code has expired.
EN_IOTA_BIND_RESULT_FAILED	255	NA	Specifies other failures.

6.3.4 IEs in EN_IOTA_LGN_IE_TYPE Messages

Enumerated Item	Value	Type	Description
EN_IOTA_LGN_IE_REASON	0	**EN_IOTA_LGN_REASON_TYPE**	Specifies the login failure reason.

6.3.5 Enumerated Values of the EN_IOTA_LGN_REASON_TYPE Parameter

Enumerated Item	Value	Type	Description
EN_IOTA_LGN_REASON_NULL	0	NA	Indicates that the reason is not specified.
EN_IOTA_LGN_REASON_CONNCET_ERR	1	NA	Specifies a connection failure.
EN_IOTA_LGN_REASON_SERVER_BUSY	2	NA	Indicates that the server is busy.
EN_IOTA_LGN_REASON_AUTH_FAILED	3	NA	Specifies an authentication failure. Developers must stop trying to reconnect the device to the IoT platform.
EN_IOTA_LGN_REASON_NET_UNAVAILABLE	5	NA	Indicates that the network is unavailable.
EN_IOTA_LGN_REASON_DEVICE_NOEXIST	12	NA	Indicates that the device does not exist. Developers must stop trying to reconnect the device to the IoT platform.
EN_IOTA_LGN_REASON_DEVICE_RMVED	13	NA	Indicates that the device does not exist. Developers must stop trying to reconnect the device to the IoT platform.
EN_IOTA_LGN_REASON_UNKNOWN	255	NA	Specifies an unknown reason.

6.3.6 Enumerated Values of the EN_IOTA_CFG_TYPE Parameter

Enumerated Item	Value	Type	Description
EN_IOTA_CFG_DEVICEID	0	String	Specifies the logical device assigned by the IoT platform.
EN_IOTA_CFG_DEVICESECRET	1	String	Specifies the authentication key for a device to access the IoT platform.
EN_IOTA_CFG_APPID	2	String	Identifies the application of a developer.
EN_IOTA_CFG_IOCM_ADDR	3	String	Specifies the IP address of the IoCM server.
EN_IOTA_CFG_IOCM_PORT	4	unsigned int	Specifies the port number of the IoCM server.
EN_IOTA_CFG_MQTT_ADDR	5	String	Specifies the IP address of the MQTT server.
EN_IOTA_CFG_MQTT_PORT	6	unsigned int	Specifies the port number of the MQTT server.

6.3.7 IEs in EN_IOTA_HUB_IE_TYPE Messages

Enumerated Item	Value	Type	Description
EN_IOTA_HUB_IE_RESULT	0	**EN_IOTA_HUB_RESULT_TYPE**	Specifies the device addition or deletion result.
EN_IOTA_HUB_IE_DEVICEID	1	String	Identifies the device assigned after successful addition.

6.3.8 Enumerated Values of the EN_IOTA_HUB_RESULT_TYPE Parameter

Enumerated Item	Value	Type	Description
EN_IOTA_HUB_RESULT_SUCCESS	0	NA	Indicates that a device is added or deleted.
EN_IOTA_HUB_RESULT_DEVICE_EXIST	1	NA	Indicates that the device already exists.
EN_IOTA_HUB_RESULT_DEVICE_NOTEXIST	2	NA	The device does not exist.
EN_IOTA_HUB_RESULT_DEVICE_FAILED	255	NA	Indicates that the execution fails.

6.3.9 IEs in EN_IOTA_DATATRANS_IE_TYPE Messages

Enumerated Item	Value	Type	Description
EN_IOTA_DATATRANS_IE_RESULT	0	unsigned int	Specifies the command execution result. The value is **0** or **1**. ● **0**: indicates that the command execution is successful. ● **1**: indicates that the command execution fails.
EN_IOTA_DATATRANS_IE_DEVICEID	1	String	Identifies a device.
EN_IOTA_DATATRANS_IE_REQUESTID	2	String	Identifies a request.
EN_IOTA_DATATRANS_IE_SERVICEID	3	String	Identifies a service.
EN_IOTA_DATATRANS_IE_METHOD	4	String	Specifies the service method.

Enumerated Item	Value	Type	Description
EN_IOTA_DATATRANS_IE_CMDCONTENT	5	String	Specifies the command content. Developers need to parse service command parameters that are assembled in JSON format according to the command definition to obtain parameter values.

6.3.10 IEs in EN_IOTA_DEVUPDATE_IE_TYPE Messages

Enumerated Item	Value	Type	Description
EN_IOTA_DEVUPDATE_IE_RESULT	0	unsigned int	Specifies the command execution result. The value is **0** or **1**. ● **0**: indicates that the command execution is successful. ● **1**: indicates that the command execution fails.
EN_IOTA_DEVUPDATE_IE_DEVICEID	1	String	Identifies a device.

6.4 Data Types

4.1 Common Data Types

Type Name	Standard Type Name
HW_INT	int
HW_UINT	unsigned int
HW_CHAR	char
HW_UCHAR	unsigned char
HW_BOOL	int
HW_ULONG	unsigned long
HW_USHORT	unsigned short

Type Name	Standard Type Name
HW_MSG	void*
HW_VOID	void
HW_NULL	0

4.2 Function Return Values

Return Value Name	Value	Standard Type Name
HW_OK	0	Execution succeeded
HW_ERR	1	Execution error
HW_ERR_PTR	2	Incorrect pointer
HW_ERR_ID	3	Incorrect ID
HW_ERR_PARA	4	Incorrect parameter
HW_ERR_KEY	5	Incorrect key
HW_ERR_NOMEM	6	Insufficient memory.
HW_ERR_MAGIC	7	Reserved
HW_ERR_OVERFLOW	8	Overflow
HW_ERR_GVAR	9	Reserved
HW_ERR_POOL	10	Reserved
HW_ERR_NO_MUTEX	11	Unlocked
HW_ERR_PID	12	Reserved
HW_ERR_FILEOPEN	13	Failed to open the file
HW_ERR_FD	14	Incorrect file description
HW_ERR_SOCKET	15	Abnormal socket
HW_ERR_NOTSUPPORT	16	Unsupported
HW_ERR_NOTLOAD	17	Not loaded
HW_ERR_ENCODE	18	Encoding error
HW_ERR_DECODE	19	Decoding error
HW_ERR_CALLBACK	22	Incorrect callback function
HW_ERR_STATE	23	Incorrect status
HW_ERR_OVERTIMES	24	Exceeded maximum number of retries

Return Value Name	Value	Standard Type Name
HW_ERR_ENDOVER	25	Reserved
HW_ERR_ENDLINE	26	Reserved
HW_ERR_NUMBER	27	Incorrect number
HW_ERR_NOMATCH	28	Mismatched
HW_ERR_NOSTART	29	Not started
HW_ERR_NOEND	30	Not ended
HW_ERR_OVERLAP	31	Reserved
HW_ERR_DROP	32	Discarded
HW_ERR_NODATA	33	No data
HW_ERR_CRC_CHK	34	CRC failed
HW_ERR_AUTH	35	Authentication failed
HW_ERR_LENGTH	36	Incorrect length
HW_ERR_NOTALLOW	37	Operation not allowed
HW_ERR_TOKEN	38	Incorrect token
HW_ERR_NOTIPV4	39	IPv4 unsupported
HW_ERR_NOTIPV6	40	IPv6 unsupported
HW_ERR_IELOST	41	Reserved
HW_ERR_IELOST1	42	Reserved
HW_ERR_IELOST2	43	Reserved
HW_ERR_AUDIO	44	Reserved
HW_ERR_VIDEO	45	Reserved
HW_ERR_MD5	46	Reserved
HW_ERR_MD5_HA1	47	Reserved
HW_ERR_MD5_RES	48	Reserved
HW_ERR_DIALOG	49	Incorrect dialog
HW_ERR_OBJ	50	Incorrect object

7 AgentLite API Reference (Java)

7.1 Before You Start

1. Overview

The IoT AgentLite (AgentLite for short) provides standard capabilities for intelligent devices to access the IoT platform in smart home, industrial IoT, connected vehicles, and other fields. These intelligent devices, such as IP cameras (IPCs), lightweight gateways, industrial gateways, and head units, have strong computing capabilities.

2. API Overview

The following table lists the functions provided by the AgentLite, the APIs for implementing these functions, and the relationship between the APIs. It helps you quickly locate APIs that are used to develop specific services.

Function	API	Description
Access of directly connected devices	BaseService.init	Initializes AgentLite resources.
	BaseService.destroy	Releases AgentLite resources.
	BindConfig.setConfig	Configures device binding.
	BindService.bind	Binds devices.
	HubService.TOPIC_UNBINDDEVICE	Receives device unbinding commands.

Function	API	Description
	LoginConfig.setConfig	Configures login information.
	LoginService.login	Connects devices to the IoT platform.
	LoginService.logout	Disconnects devices from the IoT platform.
Management of non-directly connected devices on a gateway	HubService.addDevice	Adds devices.
	HubService.updateDeviceStatus	Updates device status.
	HubService.rmvDevice	Deletes devices.
Reporting device service data	DataTransService.dataReport	Reports device service data.
Receiving device service commands	DataTransService.TOPIC_COMMAND_RECEIVE	Receives device service commands.

7.2 APIs

The external APIs provided by the AgentLite can be used for the observer mode, access of directly connected devices, management of non-directly connected devices, reporting device service data, and receiving device service commands.

7.2.1 Observer Mode

The Java-based AgentLite uses the observer mode to report data.

A third party receives data by implementing the **public void update (IotaMessage arg0)** API provided by the AgentLite.

The observer is registered by using **registerObserver(MyObserver o)**, and deregistered by using **removeObserver(MyObserver o)**.

7.2.2 Access of Directly Connected Devices

7.2.2.1 Initializing AgentLite Resources

API Function

This API is used to initialize AgentLite resources.

API Description

```
public static boolean init(String workPath, String logPath);
```


Class

BaseService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
workPath	Mandatory	String	Specifies the AgentLite working path, which is used to store the AgentLite configuration files and temporary files. The working path must be valid.
logPath	Optional	String	Specifies the log path. (If this parameter is left unspecified, logs are written to the working path.)

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Call this API to initialize AgentLite resources.
BaseService.init("/sdcard/helloWorld", null);
```

7.2.2.2 Releasing AgentLite Resources

API Function

This API is used to release all dynamic resources that are applied for, such as the memory and thread.

API Description

```
public static boolean destroy();
```

Class

BaseService

Parameter Description

N/A

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Call this API to release AgentLite resources.
BaseService.destroy();
```

7.2.2.3 Binding Configuration

API Function

This API is used to configure the IP address and port number of the IoT platform before binding a device.

API Description

```
public static boolean setConfig(int key, String value);
```

Class

BindConfig

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	int	Specifies the configuration items for device binding. ● IP address: BindConfig.BIND_CONFIG_ADDR ● Port number: BindConfig.BIND_CONFIG_PORT
value	Mandatory	String	Specifies the values of the configuration items. ● IP address: IP address of the IoT platform to which the AgentLite is connected ● Port number: 8943

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Call this API to configure the IP address and port number of the IoT platform
for device binding.
BindConfig.setConfig(BindConfig.BIND_CONFIG_ADDR, "127.0.0.1");
BindConfig.setConfig(BindConfig.BIND_CONFIG_PORT, "8943");
```

7.2.2.4 Binding a Device

API Function

A device must be bound to the IoT platform before accessing the IoT platform for the first time. The upper-layer application calls this API to transfer the device serial number, MAC address, or other device information to bind a device to the IoT platform.

Before binding a device, developers must call the **BindConfig.setConfig** API to set the IP address and port number of the IoCM server to be bound. The default port number is 8943 for the AgentLite.

API Description

```
public static boolean bind(String verifyCode, IotaDeviceInfo deviceInfo);
```

Class

BindService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
verifyCode	Mandatory	String	Specifies a device binding verification code. If the device is registered on the SP portal, set **verifyCode** to the preSecret used during device registration.
deviceInfo	Mandatory	**IotaDeviceInfo**	Specifies information about the device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

- Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the binding is successful. The binding is successful only after the **BindService** broadcast is received.
- If the binding fails, the AgentLite automatically binds the device after 30 seconds. If the retry fails for five consecutive times (the total number of attempts is six), a message is returned indicating that the binding fails and the binding stops. If developers want to re-initiate the binding, restart the device.

Example

Call this API to bind a device.

```
String verifyCode =   "123456"  ;
deviceInfo = new
IotaDeviceInfo(nodeId,manufactrueId,deviceType,model,protocolType);
BindService.bind(verifyCode,deviceInfo);
```

Implement the observer API provided by the AgentLite before calling this API.

```
//Call the bindService.registerObserver(this) to register the observer for
receiving the binding result callback and obtaining callback parameters, which
are used in login configuration.
public class AgentliteBind implements MyObserver{
    public Subscribe (Observable bindService) {
        bindService. registerObserver (this);
    }
}
//Override the update method in the AgentliteBind.
@Override
public void update(IotaMessage arg0) {
    System.out.println("BindManager receives a binding notification:" + arg0);
    int status = arg0.getUint(BindService.BIND_IE_RESULT, -1);
    System.out.println("status is :" + status);
    switch (status) {
        case 0:
        saveBindParaAndGotoLogin(arg0);
        break;
        default:
        System.out.println("============= binding failure ==============")
        bindAction();
        break;
    }
}
```

Receive a device binding response.

```
//After a device is bound, the AgentLite will return the parameters shown in the
following output. Developers must store these parameters and configure these
parameters on the device before connecting the device to the IoT platform.
//Save the parameters carried in the device binding response.
private void saveBindParaAndGotoLogin(IotaMessage iotaMsg) {
    String appId = iotaMsg.getString(BindService.BIND_IE_APPID);
    String deviceId = iotaMsg.getString(BindService.BIND_IE_DEVICEID);
```



```
    String secret = iotaMsg.getString(BindService.BIND_IE_DEVICESECRET);
    String haAddress = null, lvsAddress = null;
    saveGatewayInfo(appId, deviceId, secret, haAddress, lvsAddress);
}
```

7.2.2.5 Unbinding a Device

API Function

This API is used to passively receive the unbinding command delivered by the IoT platform. After the command is received, the configuration information of the directly connected device is deleted, and all resources are released. (If this command is received, the device has been deleted from the IoT platform.)

API Description

```
HubService.TOPIC_UNBINDDEVICE;
```

Class

HubService

Example

Implement the observer API provided by the AgentLite before calling this API.

```
public class AgentLiteHub implements MyObserver {
    public Subscribe (Observable hubService) {
        hubService. registerObserver (this);
    }
```

Receive a device unbinding response.

```
//After the device is unbound successfully, developers need to delete the
configuration information of the directly connected device and release all
resources after receiving the callback.
public void update(IotaMessage arg0) {
    // TODO Auto-generated method stub
      System.out.println("Receive a notification from HubService:" + arg0);
    int mMsgType = arg0.getMsgType();
    switch(mMsgType) {
//Receive a device adding response.
        case IodevService.IODEV_MSG_ADD_DEVICE_RSP:
        getAddDeviceAnswer(arg0);
        break;
//Receive a device deleting response.
        case IodevService.IODEV_MSG_RMV_DEVICE_RSP:
        getRmvDeviceAnswer(arg0);
        break;
//Receive a device status update response.
        case IodevService.IODEV_MSG_UPDATE_DEVSTATUS_RSP:
        getUpdateStatusAnswer(arg0);
        break;
        case IodevService.IODEV_MSG_RECEIVE_CMD:
        getUnbindAnswer(arg0);
        break;
        default:
        break;
    }
}
```

7.2.2.6 Configuring Parameters

7.2.2.6.1 Setting Service Parameters

API Function

This API is used to set the parameters required for device login.

API Description

```
public static boolean setConfig(int key, String value);
```

Class

LoginConfig

Parameter Description

Parameter	Mandatory or Optional	Type	Description
key	Mandatory	int	Specifies the configuration items for device login. ● Device ID: LoginConfig.LOGIN_CONFIG_DEVICEID ● Application ID: LoginConfig.LOGIN_CONFIG_APPID ● Secret: LoginConfig.LOGIN_CONFIG_SECRET ● HTTP address: LoginConfig.LOGIN_CONFIG_IOCM_ADDR ● HTTP port: LoginConfig.LOGIN_CONFIG_IOCM_PORT ● MQTT address: LoginConfig.LOGIN_CONFIG_MQTT_ADDR ● MQTT port: LoginConfig.LOGIN_CONFIG_MQTT_PORT
value	Mandatory	String	Specifies the values of the configuration items. ● Device ID: obtained from the callback returned after the device is bound successfully. ● Application ID: obtained from the callback returned after the device is bound successfully. ● Secret: obtained from the callback returned after the device is bound successfully. ● HTTP address: southbound access address for the AgentLite to connect to the IoT platform. ● HTTP port: 8943 ● MQTT address: southbound access address for the AgentLite to connect to the IoT platform. ● MQTT port: 8883

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Configure device login. (deviceID, secret, appID are the parameters returned
after the device is bound successfully.)
private void configLoginPara() {
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_DEVICEID,
GatewayInfo.getDeviceID());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_APPID, GatewayInfo.getAppID());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_SECRET,
GatewayInfo.getSecret());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_ADDR,
GatewayInfo.getHaAddress());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_IOCM_PORT, "8943");
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_ADDR,
GatewayInfo.getHaAddress());
    LoginConfig.setConfig(LoginConfig.LOGIN_CONFIG_MQTT_PORT, "8883");
}
```

7.2.2.6.2 (Optional) Selecting an Encryption Algorithm Type

API Function

This API is used to configure the encryption algorithm type for sensitive information before login. Developers can select an appropriate encryption algorithm type based on the security level for their services.

API Description

```
public static boolean setAlgType (int type);
```

Class

BaseService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
type	Mandatory	int	Specifies the encryption algorithm type. The value ranges from 0 to 2. ● **0**: AES 256 CBC ● **1**: AES 128 GCM ● **2**: AES 256 GCM

Return Value

Return Value	Description
true	Success
false	Failure

Example

```
//Select an encryption algorithm type.
private void configAlgPara() {
    BaseService. setAlgType (2);
}
```

7.2.2.7 Connecting a Device

API Function

This API is used to connect a device to the IoT platform after the device is bound to the IoT platform for the first time or the device restarts.

API Description

```
public static boolean login();
```

Class

LoginService

Parameter Description

N/A

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the login is successful. The login is successful only after the **LoginService** broadcast is received. Before login, the API for **Setting Service Parameters** (**LoginConfig.setConfig**) is used to transfer the required login information.

Example

Call this API to connect a device to the IoT platform.

```
LoginService.login();
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class AgentliteLogin implements MyObserver {
    public AgentliteLogin (Observable loginService) {
        loginService. registerObserver (this);
    }
//Override the update method in the AgentliteLogin.
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("LoginManager receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
            case 1:
            loginResultAction(arg0);
            break;
            case 2:
            logoutResultAction(arg0);
            break;
            default:
            break;
        }
    }
```

7.2.2.8 Disconnecting a Device

API Function

This API is used to disconnect a device from the IoT platform.

API Description

```
public static boolean logout();
```

Class

LoginService

Parameter Description

N/A

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the logout is successful. The logout is successful only after the **LoginService** broadcast is received.

Example

Call this API to disconnect a device.

```
LoginService. logout();
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class AgentliteLogin implements MyObserver {
    public AgentliteLogin (Observable loginService) {
        loginService. registerObserver (this);
    }
//Override the update method in the AgentliteLogin.
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("LoginManager receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
            case 1:
            loginResultAction(arg0);
            break;
            case 2:
            logoutResultAction(arg0);
            break;
            default:
            break;
        }
    }
```

7.2.3 Management of Non-Directly Connected Devices on a Gateway

7.2.3.1 Adding a Device

API Function

This API is used to add a device that accesses the IoT platform through the gateway. After the device is connected to the IoT platform, the IoT platform will assign a unique logical ID to the device.

API Description

```
public static boolean addDevice(int cookie, IotaDeviceInfo deviceInfo);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.

Parameter	Mandatory or Optional	Type	Description
deviceInfo	Mandatory	**IotaDeviceInfo** class	Specifies information about the device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the device is added successfully. The device is added successfully only after the **HubService** broadcast is received.

Example

Call this API to add a device.

```
HubService.addDevice(29011, new IotaDeviceInfo("nodeId", "manufacturerId",
"deviceType", "model", "protocolType"));
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class AgentliteHub implements MyObserver {
    public AgentliteHub (Observable hubService) {
        hubService. registerObserver (this);
    }
//Receive a device adding response.
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("Receive a notification from HubService:" + arg0);
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
//Receive a device adding response.
            case IodevService.IODEV_MSG_ADD_DEVICE_RSP:
            getAddDeviceAnswer(arg0);
            break;
//Receive a device deleting response.
            case IodevService.IODEV_MSG_RMV_DEVICE_RSP:
            getRmvDeviceAnswer(arg0);
            break;
//Receive a device status update response.
            case IodevService.IODEV_MSG_UPDATE_DEVSTATUS_RSP:
            getUpdateStatusAnswer(arg0);
            break;
            case IodevService.IODEV_MSG_RECEIVE_CMD:
            getUnbindAnswer(arg0);
            break;
            default:
            break;
        }
    }
```


7.2.3.2 Updating Device Status

API Function

This API is used to update status information about devices, including directly connected devices and indirectly connected devices managed by the IoT platform. The device status can be updated through this API when the device is offline or online.

The status of directly connected devices is updated based on the device login status. If a directly connected device is disconnected from the IoT platform, its status becomes offline. If a directly connected device is connected or reconnected to the IoT platform, its status becomes online and does not need to be updated through this API. Therefore, it is recommended that developers call this API to update the status of indirectly connected devices.

API Description

```
public static boolean updateDeviceStatus(int cookie, String deviceId, String
status, String statusDetail);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
deviceId	Mandatory	String	Identifies a device.
status	Mandatory	String	Specifies the device status. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline.
statusDetail	Mandatory	String	Provides the detailed device status information. ● NONE ● CONFIGURATION_PENDING ● COMMUNICATION_ERROR ● CONFIGURATION_ERROR ● BRIDGE_OFFLINE ● FIRMWARE_UPDATING ● DUTY_CYCLE ● NOT_ACTIVE

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the device status is updated successfully. The device status is updated successfully only after the **HubService** broadcast is received.

Example

Call this API to update the device status.

```
HubService.deviceStatusUpdate(0, deviceId, "ONLINE", "NONE");
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class Subscribe implements MyObserver {
    public Subscribe (Observable hubService) {
        hubService. registerObserver (this);
    }
    @Override Same as those used for adding a device.
```

7.2.3.3 Deleting a Device

API Function

This API is used to delete indirectly connected devices from the IoT platform when a new device needs to be removed from the gateway.

API Description

```
public static boolean rmvDevice(int cookie, String deviceId);
```

Class

HubService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.
deviceId	Mandatory	String	Identifies a device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the device is deleted successfully. The device is deleted successfully only after the **HubService** broadcast is received.

Example

Call this API to delete a device.

```
HubService.rmvDevice(122, deviceId);
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class Subscribe implements MyObserver {
    public Subscribe (Observable hubService) {
        hubService. registerObserver (this);
    }
    @Override
Same as those used for adding a device.
```

7.2.4 Reporting Device Service Data

API Function

This API is used by the gateway to report data of directly connected devices or indirectly connected devices to the IoT platform.

API Description

```
public static boolean dataReport(int cookie, String requstId, String deviceId,
String serviceId, String serviceProperties);
```

Class

DataTransService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
cookie	Optional	int	The value ranges from 1 to 65535.

Parameter	Mandatory or Optional	Type	Description
requstId	Mandatory	String	Identifies a request. This parameter is used to match the service command delivered by the IoT platform, which can be obtained from the broadcast described in **Receiving a Command**. ● Active data reporting: The value of this parameter is **NULL**. ● Command result reporting: The value of this parameter is the request ID of the command which the reported data matches.
deviceId	Mandatory	String	Identifies a device.
serviceId	Mandatory	String	Identifies a service, which is defined in the device profile.
serviceProperties	Mandatory	String	Specifies the service attributes, which are the parameter names and values in JSON format defined in the device profile. The attribute values are the data to be reported by the device.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the service data is reported successfully. The service data is reported successfully only after the **DataTransService** broadcast is received.

Example

Call this API to report data.

```
DataTransService.dataReport(1211, NULL, "xxxx_xxxx_xxxx_xxxx", "DoorWindow",
"{\"status\":\"OPEN\"}");
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class Subscribe implements MyObserver {
    public Subscribe (Observable dataTransService) {
        dataTransService. registerObserver (this);
```



```
    }
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("AgentLiteDataTrans receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
//Receive a data reporting response.
            case IodevService.IODEV_MSG_DATA_REPORT_RSP:
            getDataReportAnswer(arg0);
            break;
//Receive a command passively.
            case IodevService.IODEV_MSG_RECEIVE_CMD:
            getCmdReceive(arg0);
            break;
//MQTT message push
            case IodevService.IODEV_MSG_MQTT_PUB_RSP:
            //logoutResultAction(iotaMsg);
            break;
            default:
            break;
        }
    }
```

7.2.5 Receiving a Command

API Function

This API is used to process the control commands delivered by the IoT platform.

API Description

```
DataTransService.TOPIC_COMMAND_RECEIVE;
```

Class

DataTransService

Parameter Description

N/A

Return Value

N/A

Example

Implement the observer API provided by the AgentLite.

```
public class AgentLiteDataTrans implements MyObserver {
    public AgentLiteDataTrans (Observable dataTransService) {
        dataTransService. registerObserver (this);
    }
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("AgentLiteDataTrans receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
//Receive a data reporting response.
            case IodevService.IODEV_MSG_DATA_REPORT_RSP:
```



```
            getDataReportAnswer(arg0);
            break;
//Receive a command passively.
            case IodevService.IODEV_MSG_RECEIVE_CMD:
            getCmdReceive(arg0);
            break;
//MQTT message push
            case IodevService.IODEV_MSG_MQTT_PUB_RSP:
            //logoutResultAction(iotaMsg);
            break;
            default:
            break;
        }
    }
```

7.2.6 Releasing Data

API Function

This API is used by the gateway to release data of directly connected devices or indirectly connected devices to the IoT platform. In data releasing, users can select topics for the released data, which is not supported in data reporting. In addition, the JSON character strings need to be assembled by the developer and transferred to the API. The SDK is used only for transparent transmission.

API Description

```
public static boolean mqttDataPub(int cookie, String topic, int qos, byte[]
serviceData);
```

Class

DataTransService

Parameter Description

Parameter	Mandatory or Optional	Type	Description
uiCookie	Optional	int	The value ranges from 1 to 65535.
pucTopic	Mandatory	String	Specifies the topic of the released data.
uiQos	Mandatory	int	Specifies the value of **MQTT**. Default value: **1**
pbstrServiceData	Mandatory	byte[]	Specifies the packet body of the released data.

Return Value

Return Value	Description
true	Success
false	Failure

NOTE

Return values only show API calling results. For example, the return value **true** indicates that the API is called successfully but does not indicate that the service data is released successfully. The service data is released successfully only after the **DataTransService** broadcast is received.

Example

Call this API to release data.

```
DataTransService. mqttDataPub(1211, "/huawei/v1/devices/336d9bac-9ebf-44e9-95cf-efac5f05da3a/services/Storage", 1, bstrBody);
```

Implement the observer API provided by the AgentLite before calling this API.

```
public class Subscribe implements MyObserver {
    public Subscribe (Observable dataTransService) {
        dataTransService. registerObserver (this);
    }
    @Override
    public void update(IotaMessage arg0) {
        // TODO Auto-generated method stub
        System.out.println("AgentLiteDataTrans receives a notification:" + arg0)
        int mMsgType = arg0.getMsgType();
        switch(mMsgType) {
//Receive a data reporting response.
            case IodevService.IODEV_MSG_DATA_REPORT_RSP:
            getDataReportAnswer(arg0);
            break;
//Receive a command passively.
            case IodevService.IODEV_MSG_RECEIVE_CMD:
            getCmdReceive(arg0);
            break;
//MQTT message push
            case IodevService.IODEV_MSG_MQTT_PUB_RSP:
            //logoutResultAction(iotaMsg);
            break;
            case IodevService.IODEV_MSG_MQTT_SUB_RSP:
            //TopicSubcribeResultAction(iotaMsg);
            break;
            case IodevService.IODEV_MSG_MQTT_DATA_RECV_RSP:
            //DataRecvAction(iotaMsg);
            break;
            default:
            break;
        }
    }
```

7.3 Common Data Structures

7.3.1 IotaDeviceInfo Class

Parameter	Mandatory or Optional	Type	Description
nodeId	Mandatory	String	Uniquely identifies a device managed by the gateway connected to the IoT platform. This parameter is a key parameter determined by the device.
name	Optional	String	Specifies the name of a device.
description	Optional	String	Specifies the description of a device.
manufacturerId	Mandatory	String	Identifies a manufacturer.
manufacturerName	Optional	String	Specifies the name of a manufacturer.
mac	Optional	String	Specifies the MAC address of a device.
location	Optional	String	Specifies the location of a device.
deviceType	Mandatory	String	Specifies the type of a device.
model	Mandatory	String	Specifies the model of a device. ● If a directly connected device is used, the value must be the same as the model defined in the profile. ● If a Z-Wave device is used, the model is in the hexadecimal format of **ProductType** + **ProductId** (padded with zeros if required), for example, **001A-0A12**.
swVersion	Optional	String	Specifies the software version. In Z-Wave, the format is major version.minor version, for example, **1.1**.
fwVersion	Optional	String	Specifies the firmware version.
hwVersion	Optional	String	Specifies the hardware version.
protocolType	Mandatory	String	Specifies the protocol type (Z-Wave).
bridgeId	Optional	String	Specifies the bridge through which devices connect to the IoT platform.
status	Optional	String	Specifies the status of a device. The value is **ONLINE** or **OFFLINE**. ● **ONLINE**: The device is online. ● **OFFLINE**: The device is offline.

Parameter	Mandatory or Optional	Type	Description
statusDetail	Optional	String	Specifies the details about the device status. If **pcStatus** is specified, this parameter is mandatory. Value: ● NONE ● CONFIGURATION_PENDING ● COMMUNICATION_ERROR ● CONFIGURATION_ERROR ● BRIDGE_OFFLINE ● FIRMWARE_UPDATING ● DUTY_CYCLE ● NOT_ACTIVE
mute	Optional	String	Specifies whether a device is screened. ● TRUE ● FALSE

7.3.2 IotaMessage Class

```
Obtain appId, deviceId, and secret from the response returned after the device is
bound successfully.
String appId = iotaMsg.getString(BindService.BIND_IE_APPID);
String deviceId = iotaMsg.getString(BindService.BIND_IE_DEVICEID);
String secret = iotaMsg.getString(BindService.BIND_IE_DEVICESECRET);
```

```
Receive a response for data reporting.
String deviceId = iotaMsg.getString(DataTransService.DATATRANS_IE_DEVICEID);
int retcode = iotaMsg.getUint(DataTransService.DATATRANS_IE_RESULT, 0);
intcookie = iotaMsg.getUint(DataTransService.DATATRANS_IE_COOKIE, 0);
```

```
Receive a command.
String deviceId = iotaMsg.getString(DataTransService.DATATRANS_IE_DEVICEID);
String requestId = iotaMsg.getString(DataTransService.DATATRANS_IE_REQUSTID);
String serviceId = iotaMsg.getString(DataTransService.DATATRANS_IE_SERVICEID);
String method = iotaMsg.getString(DataTransService.DATATRANS_IE_METHOD);
String cmd = iotaMsg.getString(DataTransService.DATATRANS_IE_CMDCONTENT);
```

```
Receive a response for device adding. (If ret is 0, the device is added
successfully.)
String deviceId =iotaMsg.getString(HubService.HUB_IE_DEVICEID);
int ret = iotaMsg.getUint(HubService.HUB_IE_RESULT, HubService.HUB_RESULT_FAILED);
```

```
Receive a response for device deleting. (If result is 0, the device is deleted
successfully.)
int result = iotaMsg.getUint(HubService.HUB_IE_RESULT, 0);
int cookie = iotaMsg.getUint(HubService.HUB_IE_COOKIE, 0);
```

```
Receive a response for device status update. (If result is 0, the device status
is updated successfully.)
```



```
int result = iotaMsg.getUint(HubService.HUB_IE_RESULT, 0);
int cookie = iotaMsg.getUint(HubService.HUB_IE_COOKIE, 0);
```

```
Receive a response for device logout. (The error code information is defined by
referring to loginService.)
int reason = iotaMsg.getUint(LoginService.LOGIN_IE_RESULT, -1);
```

8 MQTT Interface Reference

8.1 Overview

8.1.1 Introduction to MQTT

For details about the MQTT protocol, see **mqtt-v3.1.1-os.pdf**.

NOTE

The details of MQTT syntax and APIs are subject to this protocol. Currently, only MQTTS is supported for access to the IoT platform.

8.1.2 Message Format

An MQTT message consists of fixed header, variable header, and payload.

For details about the format of fixed header and variable header, see ***MQTT Version V3.1.1***. Payload is defined by the application in the PUBLISH (PUB) message, that is, by the device and the IoT platform.

The following describes the CONNECT, SUBSCRIBE, and PUBLISH messages.

- CONNECT - Client requests a connection to a server

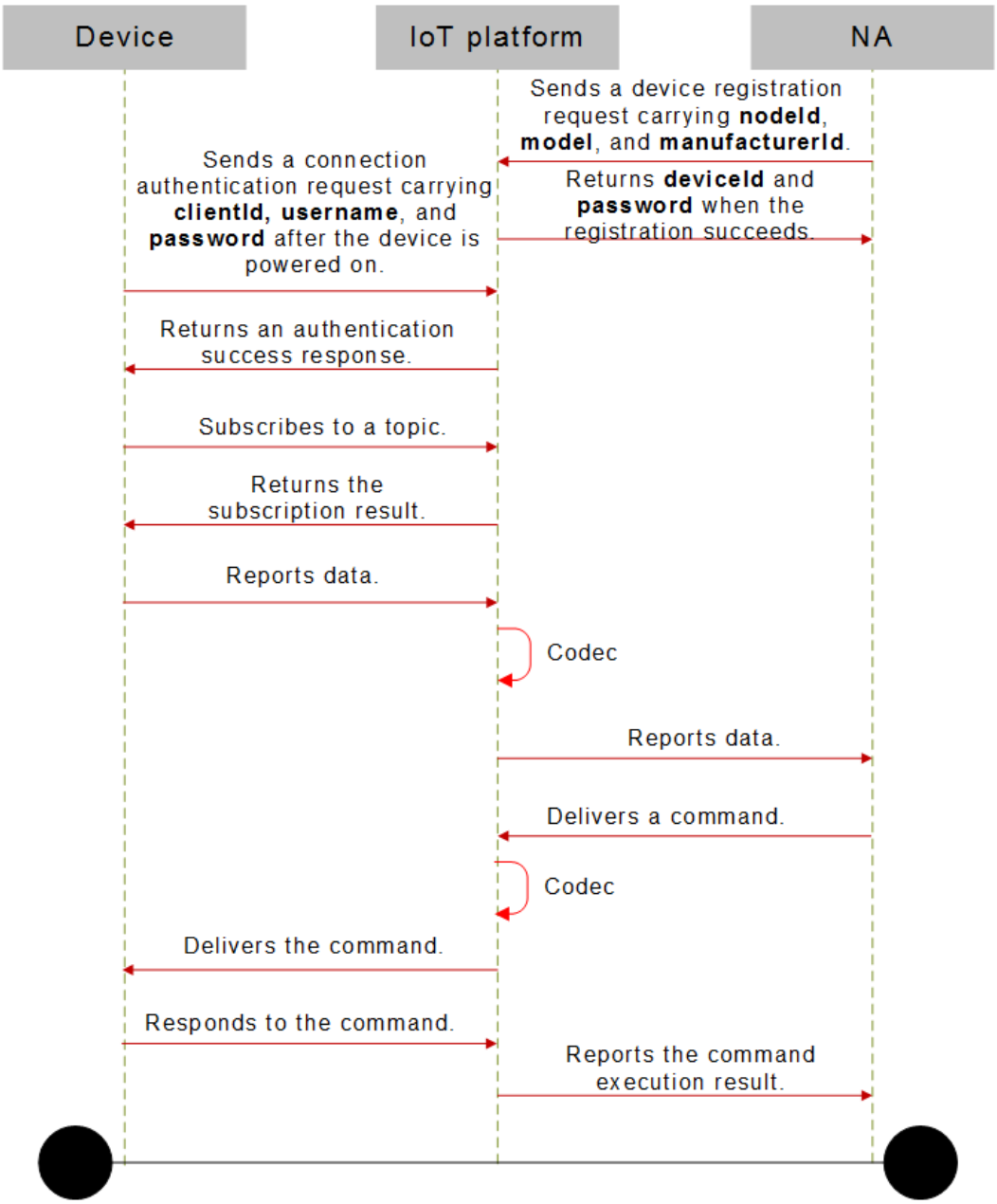
  For details about the parameters in the payload of a CONNECT message, see **MQTT Connection Authentication Through CONNECT Messages**.
- SUBSCRIBE - Subscribe to named topics

  Set **Topic name** in the payload to the topic that a device wants to subscribe to. Currently, the value of this parameter is the topic of the device. For details, see **Topic Definition**.
- PUBLISH - Publish message
  - The value of **Topic name** in the variable header is the topic of the IoT platform when the device sends a message to the IoT platform. When the device receives a message from the IoT platform, the value of **Topic name** is the topic of the device. For details, see **Topic Definition**.

- The content in the payload is a complete data reporting message or command delivery message. Currently, it is a JSON object.

8.1.3 Flow Description

8.1.3.1 Service Process (One-Device-One-Secret)



1. The application sends a device registration request to the IoT platform.
2. The registration is successful, and **deviceId** and **secret** are returned. If the registration fails, the failure cause is returned.
3. After the device is powered on, it sends a connection authentication request containing **clientId**, **username**, and **password** to the IoT platform. For details, see **MQTT Connection Authentication Through CONNECT Messages**.

4. After the authentication is successful, the IoT platform returns a success response. The device is connected to the IoT platform.
5. The device sends a SUBSCRIBE message carrying the topic that contains **deviceId** to the IoT platform.
6. The IoT platform returns the subscription result.
7. The device sends a PUBLISH message carrying data and the topic to the IoT platform.
8. The IoT platform decodes the data by using the corresponding codec plug-in.
9. The IoT platform sends the decoded data to the corresponding application based on the topic.
10. The application delivers a command to the IoT platform.
11. The IoT platform encodes the data by using the corresponding codec plug-in.
12. The IoT platform sends the encoded data to the device.
13. The device responds to the command and returns a response to the IoT platform.
14. The IoT platform sends the command execution result to the application.

    NOTE

    For details about the APIs used for the interaction between applications and the IoT platform, see the API reference document.

8.1.3.2 Device Registration (One-Device-One-Secret)

Device registration (one-device-one-secret mode) can be conducted in any of the following ways:

- Call the northbound API to register devices.

  The device registration API (**/iocm/app/reg/v2.0.0/deviceCredentials**) is used to register a device with the IoT platform in one-device-one-secret mode. After a device is successfully registered, the IoT platform will return a device secret. The **mqttConnect** field is added by this API to distinguish the devices registered in this mode.

Parameter	Mandatory or Optional	Type	Description
mqttConnect	Optional	Boolean	● **True**: indicates that the one-device-one-secret registration mode is adopted. After the registration is successful, the device ID and secret are returned. ● **False**: indicates that the original registration mode is adopted. After the registration is successful, **verifyCode** of the device is returned. When the device accesses the IoT platform for the first time, the device ID and secret are obtained by entering the verification code.

- Register a device on the management portal.

  Choose **Device Manage** > **Device** > **Registration** > **Single Registration**, click **Register**, and enter the required device information. After the device is successfully registered, **deviceId** and **secret** are returned.

8.2 API Description

8.2.1 Topic Definition

Subscription Message Topic (for Directly Connected Devices in One-Device-One-Secret Mode)

To obtain messages and data delivered by the IoT platform, a device must send an MQTT message to subscribe to a specified topic after successful login.

The format of the topic for a SUBSCRIBE message is **/huawei/v1/devices/{deviceId}/command/{codecMode}**.

- **codecMode**: indicates the encoding and decoding mode of the SUBSCRIBE messages. If a user-defined codec is used, the value of **codecMode** is **binary**. Otherwise, the value is **json**.
- **deviceId**: For one-device-one-secret devices, the value of this parameter is **deviceId** returned after the device is successfully registered when the device uses **deviceId** to access the IoT platform. The value can also be **nodeId** entered during device registration with the IoT platform when the device uses **nodeId** to access the IoT platform.

Data Reporting Message Topic (for Directly Connected Devices in One-Device-One-Secret Mode)

To report data through an MQTT channel, a device must send data to a specified topic.

The format of the topic for a PUBLISH message is **/huawei/v1/devices/{deviceId}/data/{codecMode}**.

- **codecMode**: indicates the encoding and decoding mode of the PUBLISH messages. If a user-defined codec is used, the value of **codecMode** is **binary**. Otherwise, the value is **json**.
- **deviceId**: For one-device-one-secret devices, the value of this parameter is **deviceId** returned after the device is successfully registered when the device uses **deviceId** to access the IoT platform. The value can also be **nodeId** entered during device registration with the IoT platform when the device uses **nodeId** to access the IoT platform.

8.2.2 MQTT Connection Authentication Through CONNECT Messages

Interface Function

The IoT platform provides the CONNECT message interface for MQTT devices to access the IoT platform. For details about the API specifications, see **mqtt-v3.1.1-os.pdf**. After the

device is authenticated, the MQTT connection between the device and the IoT platform is established.

NOTE

The IoT platform supports only MQTTS access. When a device establishes an MQTT connection to the IoT platform through the CONNECT message interface, the TLS certificate must be carried. Visit **Obtaining Development Resources** to obtain the TLS certificate from the **AgentLite** directory.

Parameters

Parameter	Mandatory or Optional	Type	Description
clientId	Mandatory	String(256)	The value of this parameter consists of device ID, authentication type, password signature type, and timestamp, which are separated by underscores (_). The device ID is **deviceId** returned after the device is registered successfully when the device uses **deviceId** to access the IoT platform. The value can also be **nodeId** entered during device registration with the IoT platform when the device uses **nodeId** to access the IoT platform. ● **authenticationType**: This 1-byte parameter has three value options, **0**, **1**, and **2**. **0** indicates that the device uses **deviceId** to access the IoT platform. **2** indicates that the device uses **nodeId** to access the IoT platform. **1** indicates that the one-model-one-secret mode is used for device access. ● **passwordSignatureType**: The length is 1 byte, and the value can be **0** or **1**. – **0** indicates that the timestamp is not verified using the HMACSHA256 algorithm. – **1** indicates that the timestamp is verified using the HMACSHA256 algorithm. ● **timestamp**: indicates the UTC time when the device connects to the IoT platform. The format is YYYYMMDDHH. For example, if the UTC time is 2018/7/24 17:56:20, the timestamp is 2018072417.
username	Mandatory	String(256)	● For one-device-one-secret devices, the value of this parameter is **deviceId** returned after the device is successfully registered when the device uses **deviceId** to access the IoT platform. The value can also be **nodeId** entered during device registration with the IoT platform when the device uses **nodeId** to access the IoT platform.

password	Mandatory	String(256)	The value of this parameter is the value of the device secret encrypted by using the HMACSHA256 algorithm with the timestamp as the key. The device secret is returned by the IoT platform after the successful device registration.

The IoT platform performs authentication based on the CONNECT message of the MQTT protocol. The information contained in **clientId** must be intact. When receiving a CONNECT message, the IoT platform checks the authentication type and password digest algorithm of the device.

- When the timestamp is verified using the HMACSHA256 algorithm, the IoT platform checks whether the message timestamp is consistent with the platform time and then checks whether the secret is correct.
- The timestamp must be contained in the CONNECT message even when the timestamp is not verified using the HMACSHA256 algorithm, but the IoT platform does not check whether the time is correct. In this case, only the secret is checked.

If the authentication fails, the IoT platform returns an error message and automatically disconnects the MQTT link.

8.2.3 Reporting Data

Interface Function

This interface is used by the device to report data to the IoT platform.

Parameters

Parameter	Mandatory or Optional	Type	Description
msgType	Mandatory	String	This parameter has a fixed value of **deviceReq**, which indicates that the device reports data to the IoT platform.
data	Mandatory	ServiceData[]	Data of a group of services. When batch data needs to be uploaded, you can specify this parameter. For details about the parameter structure, see the following table.

ServiceData structure

Parameter	Mandatory or Optional	Type	Description
serviceId	Mandatory	String	Service ID.
serviceData	Mandatory	ObjectNode	Data of a service. The detailed parameters are defined in the profile file.
eventTime	Optional	String	Time when the device collects data. The format is yyyyMMddTHHmmssZ, for example, **20161219T114920Z**. If this parameter is not carried in the reported data or the parameter format is incorrect, the time when the IoT platform receives the data prevails.

Example

```
MQTT
Topic: /huawei/v1/devices/{deviceId}/data/{codecMode}
Payload sent by the MQTT client:
{
"msgType":"deviceReq",
"data": [
{
"serviceId":"{serviceId}",
"serviceData":{
"meterId":"xxxx",
"dailyActivityTime":120,
"flow": "565656",
"cellId":"5656",
"signalStrength":"99",
"batteryVoltage":"3.5"
}
"eventTime":"20160503T121540Z"
} ,
{
"serviceId": "Battery",
"serviceData":  {"batteryLevel": 75},
"eventTime":"20160503T121540Z"
}
]
}
```

8.2.4 Delivering Commands

Interface Function

This interface is used by the IoT platform to deliver commands to devices.

Parameters

Parameter	Mandatory or Optional	Type	Description
msgType	Mandatory	String	This parameter has a fixed value of **cloudReq**, which indicates that the IoT platform delivers a command.
serviceId	Mandatory	String	Service ID.
cmd	Mandatory	String	Command name. For details, see the profile file.
paras	Mandatory	ObjectNode	Command parameters. Detailed fields are defined in the profile file.
mid	Mandatory	Int	2-byte unsigned command ID. It is allocated by the IoT platform and ranges from 1 to 65535. This parameter must be returned when the device is required to send a command response to the IoT platform.

Example

```
MQTT
A device can receive commands only after subscribing to the topic /huawei/v1/
devices/{deviceId}/command/{codecMode}.
The following interface is used for northbound command delivery:
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/
{serviceId}/sendCommand?appId={appId}
Payload received by the MQTT client:
{
"msgType": "{msgType}",
"serviceId": "{serviceId}",
"mid": 2016,
"cmd": "{cmd}",
"paras": {
"value": 4
}
}
```

