Database Security Service

API Reference

Issue 01

Date 2024-12-16

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property 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 Cloud 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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Calling APIs

1.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme**:

  Protocol used to transmit requests. All APIs use HTTPS.
- **Endpoint**:

  Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**.

  For example, the endpoint of IAM in region **CN-Hong Kong** is **iam.ap-southeast-1.myhuaweicloud.com**.
- **resource-path**:

  Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
- **query-string**:

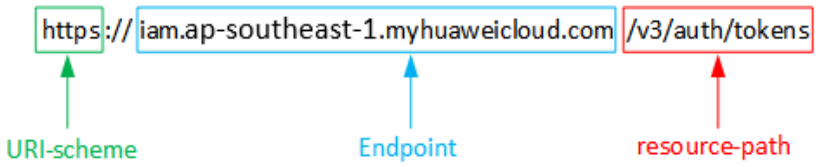
  Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint of IAM (iam.ap-southeast-1.myhuaweicloud.com)) for this region and

the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 1-1 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to **obtain a user token**, the request method is POST. The request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. The user token is a response to the API used to **obtain a user token**. This API is the only one that does not require authentication.

> **NOTE**
>
> In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.
>
> For more information, see **AK/SK-based Authentication**.

The API used to **obtain a user token** does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Set ***username*** to the name of a user, ***domainname*** to the name of the account that the user belongs to, ********* to the user's login password, and ***xxxxxxxxxxxxxxxxx*** to the project name. You can learn more information about projects from **Regions and Endpoints**. Check the value of the **Region** column.

> **NOTE**
>
> The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see **Obtaining a User Token**.

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "*********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxx"
            }
        }
```



```
    }
}
```

If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

1.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. This method is recommended because it provides higher security than token-based authentication.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

The token can be obtained by calling the required API. For more information, see **Obtaining a User Token**. A project-level token is required for calling this API, that is, **auth.scope** must be set to **project** in the request body. Example:

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:


```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see **API Signature Guide**.

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

1.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status Codes**.

For example, if status code **201** is returned for calling the API used to **obtain a user token**, the request is successful.

Response Header

A response header corresponds to a request header, for example, **Content-Type**.

Figure 1-2 shows the response header for the API of **obtaining a user token**, in which **x-subject-token** is the desired user token. Then, you can use the token to authenticate the calling of other APIs.

Figure 1-2 Header of the response to the request for obtaining a user token

(Optional) Response Body

A response body is generally returned in a structured format, corresponding to the **Content-Type** in the response header, and is used to transfer content other than the response header.

The following shows part of the response body for the API to **obtain a user token**. For the sake of space, only part of the content is displayed here.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "xxxxxxxx",
......
```

If an error occurs during API calling, the system returns an error code and a message to you. The following shows the format of an error response body:

```
{
    "error": {
        "message": "The request you have made requires authentication.",
        "title": "Unauthorized"
    }
}
```

In the preceding information, **error_code** is an error code, and **error_msg** describes the error.

2 API

2.1 Querying on the Management Side

2.1.1 Querying Account Quota Information

Function

This API is used to query tenant quota information.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/dbss/audit/quota

Table 2-1 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Request Parameter

Table 2-2 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-3 Response body parameter

Parameter	Parameter Type	Description
project_id	String	Project ID.
audit_quota	Long	Remaining quota of the audit instance
cpu	Long	Remaining CPU quota.
ram	Long	Remaining memory quota

Status code: 400

Table 2-4 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-5 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-6 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-7 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-8 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-9 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/dbss/audit/quota
```

Example Response

Status code: 200

Succeeded

```
{
  "project_id" : "0250cb8a80c24c0b9f20f557cb159aad",
  "cpu" : 796,
```



```
  "ram" : 1622016,
  "audit_quota" : 1
}
```

Status code: 400

Client errors

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ShowAuditQuotaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAuditQuotaRequest request = new ShowAuditQuotaRequest();
        try {
            ShowAuditQuotaResponse response = client.showAuditQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```



```
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAuditQuotaRequest()
        response = client.show_audit_quota(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```



```
    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAuditQuotaRequest{}
    response, err := client.ShowAuditQuota(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Client error.
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.1.2 Querying ECS Specifications

Function

This API is used to query ECS specifications.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/dbss/audit/specification

Table 2-10 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Request Parameter

Table 2-11 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-12 Response body parameter

Parameter	Parameter Type	Description
specification	Array of **EcsSpecificationBean** objects	Specifies a set of specifications.

Table 2-13 EcsSpecificationBean

Parameter	Parameter Type	Description
azs	Array of strings	AZ set to which the ECS specification belongs
id	String	ECS specification ID

Parameter	Parameter Type	Description
level	String	Specification level. The supported level depends on the site configuration. • entry: entry level • low: basic edition • medium: professional edition • high: advanced edition
name	String	Flavor name
proxy	Integer	Number of databases that can be added to the specification
ram	Integer	Memory
vcpus	Integer	VM CPU
az_type	String	AZ Type • DEDICATED • DEC • EDGE

Status code: 400

Table 2-14 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-15 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-16 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-17 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-18 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-19 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example request

```
/v1/{project_id}/dbss/audit/specification
```

Example Response

Status code: 200

Succeeded

```
{
  "specification" : [ {
    "level" : "low",
    "id" : "s2.xlarge.4",
    "name" : "s2.xlarge.4",
    "vcpus" : 4,
```



```
    "ram" : 16384,
    "proxy" : 3,
    "azs" : [ "cn-cmcc1a-01" ]
  }, {
    "level" : "medium",
    "id" : "s2.2xlarge.4",
    "name" : "s2.2xlarge.4",
    "vcpus" : 8,
    "ram" : 32768,
    "proxy" : 6,
    "azs" : [ "cn-cmcc1a-01" ]
  }, {
    "level" : "high",
    "id" : "s3.4xlarge.4",
    "name" : "s3.4xlarge.4",
    "vcpus" : 16,
    "ram" : 65536,
    "proxy" : 30,
    "azs" : [ "cn-cmcc1a-01", "cn-cmcc1b-01" ]
  } ]
}
```

Status code: 400

Client errors

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListEcsSpecificationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);
```



```
        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListEcsSpecificationRequest request = new ListEcsSpecificationRequest();
        try {
            ListEcsSpecificationResponse response = client.listEcsSpecification(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListEcsSpecificationRequest()
        response = client.list_ecs_specification(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
```



```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEcsSpecificationRequest{}
    response, err := client.ListEcsSpecification(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Client errors
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.1.3 Querying AZ Information

Function

This API is used to query AZ information.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/dbss/audit/availability-zone

Table 2-20 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Request Parameter

Table 2-21 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-22 Response body parameter

Parameter	Parameter Type	Description
azs	Array of **AzInfo** objects	AZ set

Table 2-23 AzInfo

Parameter	Parameter Type	Description
zone_name	String	AZ name
zone_number	Integer	AZ number
az_type	String	AZ type
alias	String	Chinese alias of the AZ
alias_us	String	English alias of the AZ

Status code: 400

Table 2-24 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-25 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-26 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-27 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code

Parameter	Parameter Type	Description
error_msg	String	Error message

Status code: 500

Table 2-28 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-29 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v2/{project_id}/dbss/audit/availability-zone
```

Example Response

Status code: 200

Succeeded

```
{
  "azs" : [ {
    "zone_name" : "xx-xx",
    "zone_number" : 2,
    "az_type" : "normal",
    "alias" : "AZ 2",
    "alias_us" : "AZ2"
  }, {
    "zone_name" : "xx-xx",
    "zone_number" : 1,
    "az_type" : "normal",
    "alias" : "AZ 1",
    "alias_us" : "AZ1"
  }, {
    "zone_name" : "xx-xx",
    "zone_number" : 3,
    "az_type" : "normal",
    "alias" : "AZ 3",
    "alias_us" : "AZ3"
  } ]
}
```


Status code: 400

Client errors

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAvailabilityZoneInfosSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAvailabilityZoneInfosRequest request = new ListAvailabilityZoneInfosRequest();
        try {
            ListAvailabilityZoneInfosResponse response = client.listAvailabilityZoneInfos(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAvailabilityZoneInfosRequest()
        response = client.list_availability_zone_infos(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```
        Build())

    request := &model.ListAvailabilityZoneInfosRequest{}
    response, err := client.ListAvailabilityZoneInfos(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Client errors
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.1.4 Querying User Operation Logs

Function

This API is used to query the operation logs.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{instance_id}/dbss/audit/operate-log

Table 2-30 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Parameter	Mandatory	Parameter Type	Description
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Request Parameter

Table 2-31 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-32 Request body parameter

Parameter	Mandatory	Type	Description
time	No	**TimeRangeBean** object	Query time range
user_name	No	String	Operation log username
action	No	String	Action • CREATE • DELETE • DOWNLOAD • UPDATE
result	No	String	Execution result • success • fail
page	No	String	Page number
size	No	String	Number of records on each page.

Table 2-33 TimeRangeBean

Parameter	Mandatory	Parameter Type	Description
end_time	No	String	End time. This parameter must be used together with **start_time**. The format must be *yyyy-MM-dd HH:mm:ss*. Time when an action occurred, in UTC time.
start_time	No	String	Start time. This parameter must be used together with **end_time**. The format must be *yyyy-MM-dd HH:mm:ss*. Time when an action occurred, in UTC time.
time_range	No	String	Query time range. This parameter cannot be used together with start_time and end_time. If they are used together, this parameter has a higher priority. • HALF_HOUR • HOUR • THREE_HOUR • TWELVE_HOUR • DAY • WEEK • MONTH

Response Parameters

Status code: 200

Table 2-34 Response body parameter

Parameter	Parameter Type	Description
total_num	Integer	Total number
operate_log	Array of **OperateLogInfo** objects	Operation log list

Table 2-35 OperateLogInfo

Parameter	Parameter Type	Description
id	String	Operation log ID.
user	String	Operation log username
time	String	Time when a record is generated. The format is timestamp.
action	String	Operation type of a record. The value can be: • create • update • delete • download
function	String	Function type of the record.
name	String	Operation object of a record
description	String	Description of a record
result	String	Execution result of a record. The value can be: • success • fail

Status code: 400

Table 2-36 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-37 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-38 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-39 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-40 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-41 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/operate-log

{
  "time" : {
    "time_range" : "HOUR"
  },
  "page" : 1,
  "size" : 10
}
```

Example Responses

Status code: 200

Succeeded


```
{
  "total_num" : 3,
  "operate_log" : [ {
    "id" : "1LJP-HgBCwCqSg3BVuAp",
    "user" : "hby-test",
    "time" : "2021-04-22 06:40:52",
    "function": "Database list",
    "action": "Delete",
    "name" : "db01 ",
    "description":": Delete the audited database,
    "result" : "success"
  }, {
    "id" : "07JO-HgBCwCqSg3ByOAD",
    "user" : "hby-test",
    "time" : "2021-04-22 06:40:15",
    "function": "Database list",
    "action": "Update",
    "name" : "db01 ",
    "description": "Close the audit client",
    "result" : "success"
  }, {
    "id" : "ULKM93gBCwCqSg3BZeD1",
    "user" : "hby-test",
    "time" : "2021-04-22 03:07:56",
    "function": "Database list",
    "action": "Create",
    "name" : "db01",
    "description": "Create a new database",
    "result" : "success"
  } ]
}
```

Status code: 400

Request Parameter Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal Server Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
```

```
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditOperateLogsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditOperateLogsRequest request = new ListAuditOperateLogsRequest();
        request.withInstanceId("{instance_id}");
        OperateLogGetRequest body = new OperateLogGetRequest();
        TimeRangeBean timebody = new TimeRangeBean();
        timebody.withTimeRange("HOUR");
        body.withSize("10");
        body.withPage("1");
        body.withTime(timebody);
        request.withBody(body);
        try {
            ListAuditOperateLogsResponse response = client.listAuditOperateLogs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
```



```
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditOperateLogsRequest()
        request.instance_id = "{instance_id}"
        timebody = TimeRangeBean(
            time_range="HOUR"
        )
        request.body = OperateLogGetRequest(
            size="10",
            page="1",
            time=timebody
        )
        response = client.list_audit_operate_logs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditOperateLogsRequest{}
    request.InstanceId = "{instance_id}"
    timeRangeTime:= "HOUR"
    timebody := &model.TimeRangeBean{
        TimeRange: &timeRangeTime,
    }
    sizeOperateLogGetRequest:= "10"
```



```
    pageOperateLogGetRequest:= "1"
    request.Body = &model.OperateLogGetRequest{
        Size: &sizeOperateLogGetRequest,
        Page: &pageOperateLogGetRequest,
        Time: timebody,
    }
    response, err := client.ListAuditOperateLogs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Incorrect request parameter.
403	Authentication failed.
500	Internal Server Error

Error Codes

For details, see **Error Codes**.

2.2 Audit Instance

2.2.1 Creating an Audit Instance in Yearly/Monthly Billing Mode

Function

This API is used to create an audit instance in yearly/monthly billing mode.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/dbss/audit/charge/period/order

Table 2-42 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Request Parameter

Table 2-43 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-44 Request body parameter

Parameter	Mandatory	Parameter Type	Description
name	Yes	String	Instance name. The options are as follows: • A name must contain 1 to 64 characters including digits, letters, underscores (_), and hyphens (-),
flavor_ref	Yes	String	Specifies the ID of the specifications used by the ECS.
vpc_id	Yes	String	VPC ID
availability_zone	Yes	String	Availability zone to which the ECS belongs. (Primary and secondary AZs are separated by commas. Example: az1.dc1,az2.dc2)
enterprise_project_id	No	String	Specifies the ID of the enterprise project Mandatory for interconnecting with EPS.
nics	Yes	Array of **nics** objects	NIC of the ECS

Parameter	Mandatory	Parameter Type	Description
security_groups	Yes	Array of **security_groups** objects	Information about the security group to which the ECS belongs.
comment	No	String	Indicates the remarks.
region	Yes	String	ID of the region where the ECS is located.
cloud_service_type	Yes	String	Service type: • hws.service.type.dbss
charging_mode	Yes	Integer	Billing mode: • 0: yearly/monthly • 1: pay-per-use
period_type	Yes	Integer	Subscription period type. The value can be: • **0**: day • **1**: week • **2**: month • **3**: year • **4**: hour • **5**: absolute time
period_num	Yes	Integer	Number of subscription periods
subscription_num	Yes	Integer	Number of subscriptions. Only one set of DBSS can be subscribed to.
product_infos	Yes	Array of **product_infos** objects	Product list.
tags	No	Array of **KeyValueBean** objects	Resource tag
promotion_info	No	String	Discount information
is_auto_renew	No	Integer	Auto-Renewal • 1: automatic renewal • 0: no

Table 2-45 nics

Parameter	Mandatory	Parameter Type	Description
subnet_id	Yes	String	ID of the subnet on which the NIC works.
ip_address	No	String	IP address. If the value of this parameter is left blank or is set to an empty string, an IP address is automatically assigned.

Table 2-46 security_groups

Parameter	Mandatory	Parameter Type	Description
id	Yes	String	Specifies the ID of the security group corresponding to the ECS. This ID takes effect for the NIC configured on the ECS.

Table 2-47 product_infos

Parameter	Mandatory	Parameter Type	Description
product_id	Yes	String	Product ID.
cloud_service_type	Yes	String	Service Type: • hws.service.type.dbss
resource_type	Yes	String	Resource type: • hws.resource.type.dbss
resource_spec_code	Yes	String	Resource specifications: - dbss.bypassaudit.low- dbss.bypassaudit.medium- dbss.bypassaudit.high

Parameter	Mandatory	Parameter Type	Description
product_spec_desc	No	String	Product specification description, including the host name, specifications, VPC, and subnet. JSON string format: {"specDesc":{"zh-cn":{" Host name ":"value1"," Specification ":"value2"," VPC ":"value3"," Subnet ":"value4"},"en-us":{"Instance Name":"value1","Edition":"value2","VPC":"value3","Subnet":"value4"}}}

Table 2-48 KeyValueBean

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Tag key.
value	No	String	Value

Response Parameters

Status code: 200

Table 2-49 Response body parameter

Parameter	Parameter Type	Description
description	String	Description
code	String	Returned code
order_id	String	Specifies an order ID.

Status code: 400

Table 2-50 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-51 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-52 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-53 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-54 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-55 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example request

```
/v2/{project_id}/dbss/audit/charge/period/order

{
  "flavor_ref" : "st6.xlarge.4",
  "name" : "DBSS-acc3",
  "vpc_id" : "4c035747-f77b-4c6d-b23b-cb3a2b96c7e6",
  "availability_zone" : "xx-xx",
  "comment" : "",
  "region" : "xx-xx",
  "nics" : [ {
    "subnet_id" : "6201dcf2-1374-43ec-ae8b-78b4081572d3"
  } ],
  "security_groups" : [ {
    "id" : "04088976-9c63-4e6b-9070-84e6a30c782b"
  } ],
  "cloud_service_type" : "hws.service.type.dbss",
  "charging_mode" : 0,
  "period_type" : 2,
  "period_num" : 1,
  "subscription_num" : 1,
  "is_auto_renew" : 0,
  "product_infos" : [ {
    "product_id" : "00301-xxxxxxx-0--0",
    "cloud_service_type" : "hws.service.type.dbss",
    "resource_type" : "hws.resource.type.dbss",
    "resource_spec_code" : "dbss.bypassaudit.low",
    "product_spec_desc" : "{\"specDesc\":{\"zh-cn\":{},\"en-us\":{\"instance Name\":\"DBSS-test\",\"VPC
\":\"default_vpc\",\"Subnet\":\"subnet-af32\"}}}"
  } ],
  "promotion_info" : "",
  "enterprise_project_id" : "0",
  "tags" : [ {
    "key" : "key_test",
    "value" : "1"
  } ]
}
```

Response Examples

Status code: 200

Success

```
{
  "description" : "Success",
  "code" : "0",
  "order_id" : "CS1710190909OGQIS"
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstancesPeriodOrderSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateInstancesPeriodOrderRequest request = new CreateInstancesPeriodOrderRequest();
        CreateInstancePeriodRequest body = new CreateInstancePeriodRequest();
        List<KeyValueBean> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new KeyValueBean()
                .withKey("key_test")
                .withValue("1")
        );
        List<CreateInstancePeriodRequestProductInfos> listbodyProductInfos = new ArrayList<>();
        listbodyProductInfos.add(
            new CreateInstancePeriodRequestProductInfos()
                .withProductId("00301-xxxxxxx-0--0")
                .withCloudServiceType("hws.service.type.dbss")
                .withResourceType("hws.resource.type.dbss")
                .withResourceSpecCode("dbss.bypassaudit.low")
                .withProductSpecDesc("{"specDesc":{"zh-cn":{},"en-us":{"instance Name":"DBSS-test","VPC":"default_vpc","Subnet":"subnet-af32"}}}")
        );
        List<CreateInstancePeriodRequestSecurityGroups> listbodySecurityGroups = new ArrayList<>();
        listbodySecurityGroups.add(
            new CreateInstancePeriodRequestSecurityGroups()
                .withId("04088976-9c63-4e6b-9070-84e6a30c782b")
        );
        List<CreateInstancePeriodRequestNics> listbodyNics = new ArrayList<>();
        listbodyNics.add(
            new CreateInstancePeriodRequestNics()
                .withSubnetId("6201dcf2-1374-43ec-ae8b-78b4081572d3")
        );
        body.withIsAutoRenew(0);
        body.withPromotionInfo("");
        body.withTags(listbodyTags);
        body.withProductInfos(listbodyProductInfos);
        body.withSubscriptionNum(1);
```



```
        body.withPeriodNum(1);
        body.withPeriodType(2);
        body.withChargingMode(0);
        body.withCloudServiceType("hws.service.type.dbss");
        body.withRegion("xx-xx");
        body.withComment("");
        body.withSecurityGroups(listbodySecurityGroups);
        body.withNics(listbodyNics);
        body.withEnterpriseProjectId("0");
        body.withAvailabilityZone("xx-xx");
        body.withVpcId("4c035747-f77b-4c6d-b23b-cb3a2b96c7e6");
        body.withName("DBSS-acc3");
        body.withFlavorRef("st6.xlarge.4");
        request.withBody(body);
        try {
            CreateInstancesPeriodOrderResponse response = client.createInstancesPeriodOrder(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstancesPeriodOrderRequest()
        listTagsbody = [
            KeyValueBean(
                key="key_test",
                value="1"
            )
        ]
        listProductInfosbody = [
            CreateInstancePeriodRequestProductInfos(
                product_id="00301-xxxxxxx-0--0",
```



```
            cloud_service_type="hws.service.type.dbss",
            resource_type="hws.resource.type.dbss",
            resource_spec_code="dbss.bypassaudit.low",
            product_spec_desc="{"specDesc":{"zh-cn":{},"en-us":{"instance Name":"DBSS-
test","VPC":"default_vpc","Subnet":"subnet-af32"}}}"
        )
    ]
    listSecurityGroupsbody = [
        CreateInstancePeriodRequestSecurityGroups(
            id="04088976-9c63-4e6b-9070-84e6a30c782b"
        )
    ]
    listNicsbody = [
        CreateInstancePeriodRequestNics(
            subnet_id="6201dcf2-1374-43ec-ae8b-78b4081572d3"
        )
    ]
    request.body = CreateInstancePeriodRequest(
        is_auto_renew=0,
        promotion_info="",
        tags=listTagsbody,
        product_infos=listProductInfosbody,
        subscription_num=1,
        period_num=1,
        period_type=2,
        charging_mode=0,
        cloud_service_type="hws.service.type.dbss",
        region="xx-xx",
        comment="",
        security_groups=listSecurityGroupsbody,
        nics=listNicsbody,
        enterprise_project_id="0",
        availability_zone="xx-xx",
        vpc_id="4c035747-f77b-4c6d-b23b-cb3a2b96c7e6",
        name="DBSS-acc3",
        flavor_ref="st6.xlarge.4"
    )
    response = client.create_instances_period_order(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstancesPeriodOrderRequest{}
    var listTagsbody = []model.KeyValueBean{
        {
            Key: "key_test",
            Value: "1",
        },
    }
    var listProductInfosbody = []model.CreateInstancePeriodRequestProductInfos{
        {
            ProductId: "00301-xxxxxxx-0--0",
            CloudServiceType: "hws.service.type.dbss",
            ResourceType: "hws.resource.type.dbss",
            ResourceSpecCode: "dbss.bypassaudit.low",
            ProductSpecDesc: "{"specDesc":{"zh-cn":{},"en-us":{"instance Name":"DBSS-
test","VPC":"default_vpc","Subnet":"subnet-af32"}}}",
        },
    }
    var listSecurityGroupsbody = []model.CreateInstancePeriodRequestSecurityGroups{
        {
            Id: "04088976-9c63-4e6b-9070-84e6a30c782b",
        },
    }
    var listNicsbody = []model.CreateInstancePeriodRequestNics{
        {
            SubnetId: "6201dcf2-1374-43ec-ae8b-78b4081572d3",
        },
    }
    isAutoRenewCreateInstancePeriodRequest:= int32(0)
    promotionInfoCreateInstancePeriodRequest:= ""
    commentCreateInstancePeriodRequest:= ""
    request.Body = &model.CreateInstancePeriodRequest{
        IsAutoRenew: &isAutoRenewCreateInstancePeriodRequest,
        PromotionInfo: &promotionInfoCreateInstancePeriodRequest,
        Tags: &listTagsbody,
        ProductInfos: listProductInfosbody,
        SubscriptionNum: int32(1),
        PeriodNum: int32(1),
        PeriodType: int32(2),
        ChargingMode: int32(0),
        CloudServiceType: "hws.service.type.dbss",
        Region: "xx-xx",
        Comment: &commentCreateInstancePeriodRequest,
        SecurityGroups: listSecurityGroupsbody,
        Nics: listNicsbody,
        EnterpriseProjectId: "0",
        AvailabilityZone: "xx-xx",
        VpcId: "4c035747-f77b-4c6d-b23b-cb3a2b96c7e6",
        Name: "DBSS-acc3",
        FlavorRef: "st6.xlarge.4",
    }
    response, err := client.CreateInstancesPeriodOrder(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.2.2 Querying Information About an Instance Creation Task

Function

This API is used to query information about an instance creation task.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/dbss/audit/jobs/{resource_id}

Table 2-56 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_id	Yes	String	Resource ID You can obtain the value from resource_id in the API for querying the instance list.

Request Parameter

Table 2-57 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-58 Response body parameter

Parameter	Parameter Type	Description
jobs	Array of **JobBean** objects	Table for tasks of creating instances

Table 2-59 JobBean

Parameter	Parameter Type	Description
job_id	String	Task ID
status	String	Task status. • SUCCESS • RUNNING • FAIL • INIT • READY
job_type	String	Type
server_id	String	VM ID
server_name	String	VM Name
resource_id	String	Resource ID
begin_time	Long	Start time

Parameter	Parameter Type	Description
end_time	Long	End time
charge_mode	String	Billing mode • Period: period-based charging • Demand: pay-per-use
error_code	String	Error code
fail_reason	String	Failure cause.
ha_id	String	Protected instance ID. This field has been discarded.
ha_name	String	Protected instance name. This field has been discarded.

Status code: 400

Table 2-60 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-61 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-62 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-63 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-64 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-65 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/dbss/audit/jobs/{resource_id}
```

Example Response

Status code: 200

Succeeded

```
{
  "jobs": [
    {
      "resource_id": "2c154fdd-0d43-47b7-9cf1-5236bf6a2ca7",
      "status": "SUCCESS",
      "job_type": null,
      "job_id": "8abf9647852a1daa01852e517e1a1a0b",
      "begin_time": 1671519371000,
      "end_time": 1671519417000,
      "error_code": null,
      "fail_reason": null,
      "charge_mode": "Demand",
      "server_name": "DBSS-qct-1220",
      "server_id": "0aa8f621-bc19-4822-b66d-7ab9ae3c8693",
      "ha_id": null,
      "ha_name": null
    }
```



```
 ]
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal Server Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditInstanceJobsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditInstanceJobsRequest request = new ListAuditInstanceJobsRequest();
        request.withResourceId("{resource_id}");
        try {
            ListAuditInstanceJobsResponse response = client.listAuditInstanceJobs(request);
```



```
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditInstanceJobsRequest()
        request.resource_id = "{resource_id}"
        response = client.list_audit_instance_jobs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditInstanceJobsRequest{}
    request.ResourceId = "{resource_id}"
    response, err := client.ListAuditInstanceJobs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Failed
403	Authentication failed.
500	Internal server error.

Error Codes

For details, see **Error Codes**.

2.2.3 Querying the Audit Instance List

Function

This API is used to query the audit instance list.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/dbss/audit/instances

Table 2-66 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.

Table 2-67 Query parameter

Parameter	Mandatory	Parameter Type	Description
offset	No	String	Offset
limit	No	String	Number of query records.

Request Parameter

Table 2-68 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-69 Response body parameter

Parameter	Parameter Type	Description
servers	Array of **AuditInstanceListBean** objects	Querying the instance information list
total	Integer	Total number

Table 2-70 AuditInstanceListBean

Parameter	Parameter Type	Description
charge_model	String	Billing mode • Period • Demand
comment	String	Remarks
config_num	Integer	Total number of databases configured for the current instance.
connect_ip	String	IPv4 address of the tenant NIC attached to the instance.
connect_ipv6	String	IPv6 address of the tenant NIC attached to the instance.
cpu	Integer	CPUs
created	String	Creation time
database_limit	Integer	Number of supported databases
effect	Integer	Instance frozen status. The value can be: • 1: The instance is frozen and can be released. • 2: The instance is frozen and cannot be released. • 3: The instance is frozen and cannot be renewed.
expired	String	Expiration time.
id	String	ID
keep_days	String	Remaining period (days)
name	String	Instance alias

Parameter	Parameter Type	Description
new_version	String	If a value is returned, upgrade is required. If no value is returned, the value is null.
port_id	String	ID of the port to which the EIP is bound
ram	Integer	Memory
region	String	Region where the instance is located.
remain_days	String	Days to expiry
resource_id	String	Resource ID.
resource_spec_code	String	Instance specification ID
scene	String	Scenario
security_group_id	String	Security Group
specification	String	Specification Type
status	String	Instance status. • SHUTOFF: The instance is shut down. • ACTIVE: The instance is running. • DELETING: The instance is being deleted. No operations are allowed. • BUILD: The instance is being created. No operations are allowed. • DELETED: The instance is deleted and does not need to be displayed. • ERROR: The instance is faulty. The instance can only be deleted. No other operations are allowed. • HAWAIT: The standby is being created. No operations are allowed. • FROZEN: The instance is frozen. Only renewal, binding, and unbinding operations are allowed. • UPGRADING: The instance is being upgraded. New upgrade operations are not allowed.
subnet_id	String	Subnet ID

Parameter	Parameter Type	Description
task	String	Task status. • powering-on: The instance is being started and can be bound to or unbound. • powering-off: The instance is being shut down and can be bound to or unbound. • rebooting: The instance is being restarted and can be bound to or unbound. • delete_wait: The instance is waiting for deletion. No operations are allowed on the cluster or the instance. • NO_TASK: The task is not displayed.
version	String	Current instance version
vpc_id	String	VPC
zone	String	AZ

Status code: 400

Table 2-71 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-72 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-73 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-74 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-75 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-76 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example request

/v1/{project_id}/dbss/audit/instances

Example Response

Status code: 200

Succeeded


```
{
  "servers" : [ {
    "name" : "DBSS-Test",
    "comment" : "",
    "connect_ipv6" : null,
    "status" : "ACTIVE",
    "task" : "NO_TASK",
    "id" : "8c53ed03-8ed7-4ff2-ad97-7b2d6d1dd364",
    "specification" : "Low | 3 Proxy",
    "zone" : "cn-cmcc1a-01",
    "created" : "2021-04-21 04:37:54",
    "expired" : null,
    "subnet_id" : "97ef0bb5-3759-4db4-aa49-0d087ed49ce5",
    "cpu" : 4,
```



```
    "ram" : 16384,
    "region" : "cn-cmcc1",
    "version" : "21.04.16.164614",
    "charge_model" : "Demand",
    "remain_days" : null,
    "config_num" : 1,
    "effect" : null,
    "scene" : null,
    "connect_ip" : "192.168.0.229",
    "port_id" : "dc4bd420-e01c-4d12-a7ff-814f17c63079",
    "resource_id" : "062212d8-8e30-4783-9671-43f3f1f3bb1e",
    "vpc_id" : "76d98391-5abc-46ed-b8a8-f664202cb166",
    "security_group_id" : "f0fbec06-bcf6-4c7e-99fa-f0ddfbb1d9bd",
    "resource_spec_code" : "dbss.bypassaudit.low",
    "keep_days" : null,
    "new_version" : null,
    "database_limit" : 3
  } ],
  "total" : 1
}
```


Status code: 400

Failed


```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```


Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
```

```
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditInstancesRequest request = new ListAuditInstancesRequest();
        try {
            ListAuditInstancesResponse response = client.listAuditInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditInstancesRequest()
        response = client.list_audit_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)
```



```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditInstancesRequest{}
    response, err := client.ListAuditInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.3 Auditing a Database

2.3.1 Querying the Database List

Function

This API is used to query the database list.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{instance_id}/dbss/audit/databases

Table 2-77 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Table 2-78 Query parameter

Parameter	Mandatory	Parameter Type	Description
status	No	String	Instance status. • ON • OFF
offset	No	String	Offset. The query starts after the first data record offsets the number of data records. The default value is 0.
limit	No	String	Number of query records. The default value of this parameter is **100**.

Request Parameter

Table 2-79 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-80 Response body parameter

Parameter	Parameter Type	Description
databases	Array of **DataBaseBean** objects	Database list
total	Integer	Total number

Table 2-81 DataBaseBean

Parameter	Parameter Type	Description
database	**DataBase** object	Database Information

Table 2-82 Database

Parameter	Type	Description
id	String	Database ID.
name	String	Database Name

Parameter	Type	Description
type	String	Type of the database to be added. The value can be: • MYSQL • ORACLE • POSTGRESQL • SQLSERVER • DAMENG • TAURUS • DWS • KINGBASE • GAUSSDBOPENGAUSS • GREENPLUM • HIGHGO • SHENTONG • GBASE8A • GBASE8S • GBASEXDM • MONGODB • DDS
version	String	DB version.
charset	String	Database character set • GBK • UTF8
ip	String	Database IP Address
port	String	Database Port
os	String	Database OS
status	String	Database status. The value can be: • ON: enabled • OFF: disabled
instance_name	String	DB instance name
audit_status	String	Database status. The value can be: • ACTIVE • SHUTOFF • ERROR
agent_url	Array of strings	Unique ID of an agent

Parameter	Type	Description
db_classification	String	Database classification. The value can be: • RDS: RDS database • ECS: self-built database
rds_audit_switch_mismatch	Boolean	The audit switch status of the RDS instance does not match. When the database audit function is enabled and the log upload function on RDS is disabled, the value of this field is true.
rds_id	String	ID of the RDS database.
rds_obj_info	String	RDS database information.
dws_obj_info	String	GaussDB(DWS) database information.
clouddb_obj_info	String	Cloud database information. This field has been discarded.

Status code: 400

Table 2-83 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-84 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-85 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-86 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-87 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-88 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/databases
```

Example Response

Status code: 200

Succeeded

```
{
  "databases" : [ {
    "database" : {
      "id" : "zLKv83gBCwCqSg3BJt0m",
      "name" : "db01",
      "type" : "MYSQL",
      "version" : "5.0",
      "charset" : "UTF8",
      "ip" : "192.168.0.204",
      "port" : "3306",
      "os" : "LINUX64",
      "status" : "OFF",
      "instance_name" : "",
      "audit_status" : null,
      "agent_url" : [ "zrKw83gBCwCqSg3Bkt1P" ],
      "db_classification" : "ECS",
    }
  } ]
```



```
  "total": 1
}
```

Status code: 400

Incorrect request parameter.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal server error.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditDatabasesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditDatabasesRequest request = new ListAuditDatabasesRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListAuditDatabasesResponse response = client.listAuditDatabases(request);
```

```
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditDatabasesRequest()
        request.instance_id = "{instance_id}"
        response = client.list_audit_databases(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditDatabasesRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListAuditDatabases(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Succeeded
400	Incorrect request parameter.
403	Authentication failed.
500	Internal server error.

Error Codes

For details, see **Error Codes**.

2.3.2 Adding an RDS Database

Function

This API is used to add an RDS database.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/{instance_id}/audit/databases/rds

Table 2-89 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Request Parameter

Table 2-90 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-91 Request body parameter

Parameter	Mandatory	Parameter Type	Description
databases	Yes	Array of **databases** objects	Database list

Table 2-92 Databases

Parameter	Mandatory	Parameter Type	Description
id	Yes	String	RDS database ID, which can be obtained from the ID field of the API for querying the RDS database list.
type	Yes	String	Database Type • MYSQL • ORACLE • POSTGRESQL • SQLSERVER • DAMENG • TAURUS • DWS • KINGBASE • MARIADB • GAUSSDBOPENGAUSS

Response Parameters

Status code: 200

Table 2-93 Response body parameter

Parameter	Parameter Type	Description
ret_list	Array of **ret_list** objects	Result list.

Table 2-94 ret_list

Parameter	Parameter Type	Description
id	String	RDS database ID
ret_status	String	Status • SUCCESS • FAILED
ret_message	String	Description

Status code: 400

Table 2-95 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-96 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-97 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-98 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-99 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-100 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v2/{project_id}/{instance_id}/audit/databases/rds

{
  "databases" : [ {
    "id" : "123751d3ee2f47aea64822e98318c6a8in01",
    "type" : "MYSQL"
  } ]
}
```

Example Responses

Status code: 200

Succeeded

```
{
  "ret_list" : [ {
    "id" : "123751d3ee2f47aea64822e98318c6a8in01",
    "ret_status" : "SUCCESS",
    "ret_message" : null
  }, {
    "id" : "2343f7285d684fed8b09fac201c3fc7ain01",
    "ret_status" : "FAILED",
    "ret_message" : "Unknown error."
  } ]
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status Code

Status Code	Description
200	Success
400	Failed
403	Authentication failed.

Status Code	Description
500	Server error.

Error Codes

For details, see **Error Codes**.

2.4 Auditing Agent

2.4.1 Enabling or Disabling an Agent

Function

This API is used to enable or disable the agent audit function. An enabled agent collects user access information.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{instance_id}/audit/agent/switch

Table 2-101 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Request Parameter

Table 2-102 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 2-103 Request body parameter

Parameter	Mandatory	Parameter Type	Description
agent_id	Yes	String	Audit agent ID. You can obtain the value from the ID field of the API for querying the agent list.
status	Yes	Integer	Agent status. The value can be: • **1**: enabled. • **0**: disabled.

Response Parameters

Status code: 200

Table 2-104 Response body parameter

Parameter	Parameter Type	Description
result	String	Response status.

Status code: 400

Table 2-105 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-106 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-107 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-108 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/audit/agent/switch

{
  "agent_id" : "ASWDSDSDSWEWDSDSD",
  "status" : 1
}
```

Response Examples

Status code: 200

The request has succeeded.

```
{
  "result" : "SUCCESS"
}
```

Status code: 400

Invalid request parameters.

```
{
  "error" : {
```



```
    "error_code" : "DBSS.XXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 403

Authentication failed.

```
{
  "error" : {
    "error_code" : "DBSS.XXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class SwitchAgentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        SwitchAgentRequest request = new SwitchAgentRequest();
        request.withInstanceId("{instance_id}");
        AgentSwitchRequest body = new AgentSwitchRequest();
        body.withStatus(1);
        body.withAgentId("ASWDSDSDSWEWDSDSD");
        request.withBody(body);
        try {
            SwitchAgentResponse response = client.switchAgent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
```



```
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SwitchAgentRequest()
        request.instance_id = "{instance_id}"
        request.body = AgentSwitchRequest(
            status=1,
            agent_id="ASWDSDSDSWEWDSDSD"
        )
        response = client.switch_agent(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchAgentRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.AgentSwitchRequest{
        Status: int32(1),
        AgentId: "ASWDSDSDSWEWDSDSD",
    }
    response, err := client.SwitchAgent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	The request is processed.
400	Invalid request parameters.
403	Authentication failed.

Error Codes

For details, see **Error Codes**.

2.5 Audit Rules

2.5.1 Enabling or Disabling a Risk Rule

Function

This API is used to enable or disable risk rules.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{instance_id}/audit/rule/risk/switch

Table 2-109 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Request Parameter

Table 2-110 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 2-111 Request body parameter

Parameter	Mandatory	Parameter Type	Description
ids	Yes	String	Rule ID. Use commas (,) to separate multiple IDs. You can obtain the value from the ID field in the API for querying risk rule policies.
status	Yes	String	Switch status • OFF: disabled • ON: enabled

Response Parameters

Status code: 200

Table 2-112 Response body parameter

Parameter	Parameter Type	Description
status	String	Response status

Status code: 400

Table 2-113 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-114 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-115 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-116 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/audit/rule/risk/switch

{
  "ids" : "c7ILB3kBCwCqSg3B2OpF",
  "status" : "OFF"
}
```

Response Examples

Example response with status code **200**:

The request has succeeded.

```
{
  "status" : "SUCCESS"
}
```

Status code: 400

Invalid request parameters.

```
{
  "error" : {
    "error_code" : "DBSS.XXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 403

Authentication failed.

```
{
  "error" : {
    "error_code" : "DBSS.XXX",
    "error_msg" : "XXX"
  }
}
```


Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class SwitchRiskRuleSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        SwitchRiskRuleRequest request = new SwitchRiskRuleRequest();
        request.withInstanceId("{instance_id}");
        BatchSwitchesRequest body = new BatchSwitchesRequest();
        body.withStatus(BatchSwitchesRequest.StatusEnum.fromValue("OFF"));
        body.withIds("c7ILB3kBCwCqSg3B2OpF");
        request.withBody(body);
        try {
            SwitchRiskRuleResponse response = client.switchRiskRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SwitchRiskRuleRequest()
        request.instance_id = "{instance_id}"
        request.body = BatchSwitchesRequest(
            status="OFF",
            ids="c7ILB3kBCwCqSg3B2OpF"
        )
        response = client.switch_risk_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
    request := &model.SwitchRiskRuleRequest{}
    request.InstanceId = "{instance_id}"
    statusBatchSwitchesRequest:= model.GetBatchSwitchesRequestStatusEnum().OFF
    idsBatchSwitchesRequest:= "c7ILB3kBCwCqSg3B2OpF"
    request.Body = &model.BatchSwitchesRequest{
        Status: &statusBatchSwitchesRequest,
        Ids: &idsBatchSwitchesRequest,
    }
    response, err := client.SwitchRiskRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
403	Authentication failed.

Error Codes

For details, see **Error Codes**.

2.5.2 Querying the Policy List of an Audit Scope

Function

This API is used to query the audit scope policy list.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{instance_id}/dbss/audit/rule/scopes

Table 2-117 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Table 2-118 Query parameter

Parameter	Mandatory	Parameter Type	Description
offset	No	String	Offset
limit	No	String	Number of query records.

Request Parameter

Table 2-119 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-120 Response body parameter

Parameter	Parameter Type	Description
scopes	Array of **RuleScopeInfo** objects	Audit scope rule list
total	Integer	Total number

Table 2-121 RuleScopeInfo

Parameter	Parameter Type	Description
id	String	Audit scope rule ID
name	String	Audit scope name
action	String	Actions in the audit scope
status	String	Audit scope rule status
exception_ips	String	Exception IP address of the audit scope
source_ips	String	Source IP address of the audit scope rule
source_ports	String	Port of the audit scope rule
db_ids	String	Database ID
db_names	String	Database name
db_users	String	Database user
all_audit	Boolean	Indicates if it is a full audit policy. The value is **TRUE** for the built-in full audit rule and **FALSE** for the user-defined full audit rule.

Status code: 400

Table 2-122 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-123 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-124 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-125 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-126 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-127 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/rule/scopes
```

Example Response

Status code: 200

Succeeded

```
{
  "scopes" : [ {
    "id" : "zX4W2ngBo47GiyUSBuNs",
    "name": "Full Audit Rule,
    "action" : "",
    "status" : "ON",
```



```
    "exception_ips" : "",
    "source_ips" : "",
    "source_ports" : "",
    "db_ids" : "",
    "db_names" : "",
    "db_users" : "",
    "all_audit" : true
  } ],
  "total" : 1
}
```

Status code: 400

Request Parameter Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal Server Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditRuleScopesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);
```



```
        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditRuleScopesRequest request = new ListAuditRuleScopesRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListAuditRuleScopesResponse response = client.listAuditRuleScopes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditRuleScopesRequest()
        request.instance_id = "{instance_id}"
        response = client.list_audit_rule_scopes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
```



```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditRuleScopesRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListAuditRuleScopes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Incorrect request parameter.
403	Authentication failed.
500	Internal server error.

Error Codes

For details, see **Error Codes**.

2.5.3 Querying SQL Injection Rule Policies

Function

This API is used to query an SQL injection rule policy.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{instance_id}/dbss/audit/rule/sql-injections

Table 2-128 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Request Parameter

Table 2-129 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-130 Request body parameter

Parameter	Mandatory	Parameter Type	Description
risk_levels	No	String	Risk severity. • HIGH • MEDIUM • LOW • NO_RISK

Response Parameters

Status code: 200

Table 2-131 Response body parameter

Parameter	Parameter Type	Description
rules	Array of **rules** objects	SQL rule list
total	Integer	Total number

Table 2-132 rules

Parameter	Parameter Type	Description
id	String	SQL rule ID
name	String	SQL rule name
status	String	Rule status. The options are as follows: • ON • OFF
risk_level	String	Risk severity. • HIGH • MEDIUM • LOW
type	String	Rule types. The value can be: • SYSTEM: built-in rules • CUSTOMIZE: user-defined rules

Parameter	Parameter Type	Description
rank	Integer	Priority. A smaller value indicates a higher priority.
feature	String	SQL command features
regex	String	Regular expression

Status code: 400

Table 2-133 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-134 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-135 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-136 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-137 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-138 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/rule/sql-injections

{
  "risk_levels" : "HIGH"
}
```

Response Examples

Status code: 200

Succeeded

```
{
  "rules" : [ {
    "id" : "zX4W2ngBo47GiyUSBuNs",
    "name": "MySQL error based SQL injection",
    "status" : "ON",
    "type" : "SYSTEM",
    "risk_level" : "HIGH",
    "rank" : 1,
    "feature": "regular expression,
    "regex" : "((.*)?(select)\\s+[0-9]+\\s+from\\s+\\(\\s*select\\s+count(.*)?(concat)\\s*(.*)?(from)\
\s*(information_schema.tables)(.*)?(group)\\s+(by)(.*)?)"
  } ],
  "total" : 1
}
```

Status code: 400

Incorrect request parameters.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal server error.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListSqlInjectionRulesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSqlInjectionRulesRequest request = new ListSqlInjectionRulesRequest();
        request.withInstanceId("{instance_id}");
        SqlRuleRequest body = new SqlRuleRequest();
        body.withRiskLevels("HIGH");
        request.withBody(body);
        try {
            ListSqlInjectionRulesResponse response = client.listSqlInjectionRules(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSqlInjectionRulesRequest()
        request.instance_id = "{instance_id}"
        request.body = SqlRuleRequest(
            risk_levels="HIGH"
        )
        response = client.list_sql_injection_rules(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```



```
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSqlInjectionRulesRequest{}
    request.InstanceId = "{instance_id}"
    riskLevelsSqlRuleRequest:= "HIGH"
    request.Body = &model.SqlRuleRequest{
        RiskLevels: &riskLevelsSqlRuleRequest,
    }
    response, err := client.ListSqlInjectionRules(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Incorrect request parameter.
403	Authentication failed.
500	Internal server error.

Error Codes

For details, see **Error Codes**.

2.5.4 Querying a Risk Rule Policy

Function

This API is used to query a risk rule policy.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{instance_id}/dbss/audit/rule/risk

Table 2-139 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Table 2-140 Query parameter

Parameter	Mandatory	Type	Description
name	No	String	Risk name
risk_levels	No	String	Risk severity. • LOW • MEDIUM • HIGH • NO_RISK

Request Parameter

Table 2-141 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-142 Response body parameter

Parameter	Parameter Type	Description
rules	Array of **rules** objects	List of risk rules
total	Integer	Total number

Table 2-143 rules

Parameter	Parameter Type	Description
id	String	Risk rule ID
name	String	Risk rule name
type	String	Risk rule type. The options are as follows: • LOGIN • OPERATE
feature	String	Risk rule characteristics
status	String	Risk rule status. The value can be: • ON: enabled • OFF: disabled
rank	Integer	Risk rule priority. A smaller value indicates a higher priority.
risk_level	String	Risk severity. • LOW • MEDIUM • HIGH • NO_RISK]
rule_type	String	Rule types. The value can be: • SYSTEM: system rule • NULL: user-defined rules

Status code: 400

Table 2-144 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-145 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-146 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-147 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-148 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-149 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/rule/risk
```

Example Response

Status code: 200

Succeeded

```
{
  "rules" : [ {
    "id" : "xX4W2ngBo47GiyUSBeOy",
    "name" : "Database_drag_detection",
    "type" : "OPERATE",
    "feature" : "CLIENT[Any]OPERATE[[SELECT]OBJECT[Any]",
    "status" : "ON",
    "rank" : -1,
    "rule_type" : "CUSTOMIZE",
    "risk_level" : "HIGH"
  }, {
    "id" : "xn4W2ngBo47GiyUSBeP4",
    "name" : "Database_Slow_SQL_Detection",
    "type" : "OPERATE",
    "feature" : "CLIENT[Any]OPERATE[[SELECT]OBJECT[Any]",
    "status" : "ON",
    "rank" : -2,
    "rule_type" : "CUSTOMIZE",
    "risk_level" : "LOW"
  } ],
  "total" : 2
}
```

Status code: 400

Request Parameter Error

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal server error.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```


Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditRuleRisksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditRuleRisksRequest request = new ListAuditRuleRisksRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListAuditRuleRisksResponse response = client.listAuditRuleRisks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *
```



```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditRuleRisksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_audit_rule_risks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditRuleRisksRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListAuditRuleRisks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```



```
  }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Incorrect request parameter.
403	Authentication failed.
500	Internal server error.

Error Codes

For details, see **Error Codes**.

2.5.5 Querying a Specified Risk Rule Policy

Function

This API is used to query a specified risk rule policy.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{instance_id}/dbss/audit/rule/risk/{rule_id}

Table 2-150 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Parameter	Mandatory	Parameter Type	Description
rule_id	Yes	String	Risk rule ID. You can obtain the value from the ID field in the API for querying risk rule policies.

Request Parameter

Table 2-151 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-152 Response body parameter

Parameter	Parameter Type	Description
rule_id	String	Risk rule ID
rule_name	String	Risk rule name
status	String	Risk rule status. The value can be: • OFF • ON
action	String	Operation set, which is separated by commas (,). LOGIN,CREATE_TABLE,CREATE_TABLESPACE,DROP_TABLE, DROP_TABLESPACE,DELETE,INSERT,INSERT_SELECT,SELECT,SELECT_FOR_UPDATE, UPDATE,CREATE_USER,DROP_USER,GRANT,OPERATE ALL

Parameter	Parameter Type	Description
schemas	Array of **schemas** objects	Schemas
rank	Integer	Risk rule priority. A smaller value indicates a higher priority.
ignore_case	Boolean	Whether to ignore the case of the customized operation object
risk_level	String	Risk severity. • LOW • MEDIUM • HIGH • NO_RISK
db_ids	String	Database ID. Values are separated by commas (,). A single ID can contain up to 256 characters.
execution_symbol	String	Relationship between the execution duration and the execution duration threshold. The value can be: • GREATER • EQUAL • LESS • GREATER_EQUAL • LESS_EQUAL • NO_MATCH
execution_time	Integer	Execution duration threshold
affect_symbol	String	Relationship between the number of affected rows and the rows threshold. The value can be: • GREATER • EQUAL • LESS • GREATER_EQUAL • LESS_EQUAL • NO_MATCH
affect_rows	Integer	Threshold of affected rows
client_ips	String	Client IP address segment. The value is in the IP-IP format or IP/*XX* format. IP address segments are separated by commas (,).

Table 2-153 schemas

Parameter	Parameter Type	Description
schema	String	Schema name
table	String	Table
column	String	Column

Status code: 400

Table 2-154 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-155 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-156 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-157 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code

Parameter	Parameter Type	Description
error_msg	String	Error message

Status code: 500

Table 2-158 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-159 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example request

```
/v1/{project_id}/{instance_id}/dbss/audit/rule/risk/{rule_id}
```

Example Response

Status code: 200

Execution succeeded.

```
{
  "status" : "OFF",
  "action" : "LOGIN,SELECT,INSERT",
  "schemas" : [ {
    "schema" : "dbss_audit",
    "table" : null,
    "column" : null
  } ],
  "rank" : 6,
  "ignore_case" : false,
  "rule_id" : "AWT0HznX7At9UslqwTfm",
  "rule_name" : "risk_rule_name_00",
  "risk_level" : "MEDIUM",
  "db_ids" : "11111,22222",
  "execution_symbol" : "GREATER",
  "execution_time" : 10000,
  "affect_symbol" : "GREATER",
  "affect_rows" : 30,
  "client_ips" : "192.168.0.1"
}
```

Status code: 400

Incorrect request parameter.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Status code: 500

Internal server error.

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ShowAuditRuleRiskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAuditRuleRiskRequest request = new ShowAuditRuleRiskRequest();
        request.withInstanceId("{instance_id}");
        request.withRuleId("{rule_id}");
        try {
            ShowAuditRuleRiskResponse response = client.showAuditRuleRisk(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



```
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAuditRuleRiskRequest()
        request.instance_id = "{instance_id}"
        request.rule_id = "{risk_id}"
        response = client.show_audit_rule_risk(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```



```
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAuditRuleRiskRequest{}
    request.InstanceId = "{instance_id}"
    request.RuleId = "{rule_id}"
    response, err := client.ShowAuditRuleRisk(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Incorrect request parameter.
403	Authentication failed.
500	Internal Server Error

Error Codes

For details, see **Error Codes**.

2.5.6 Querying a Privacy Data Masking Rule

Function

This API is used to query a privacy data masking rule.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{instance_id}/dbss/audit/sensitive/masks

Table 2-160 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID. You can obtain the value from the ID field in the API for querying the instance list.

Table 2-161 Query parameter

Parameter	Mandatory	Parameter Type	Description
offset	No	String	Offset
limit	No	String	Number of query records.

Request Parameter

Table 2-162 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-163 Response body parameter

Parameter	Parameter Type	Description
rules	Array of **rules** objects	Rules
total	Integer	Total number

Table 2-164 rules

Parameter	Parameter Type	Description
id	String	Rule ID
name	String	Rule name.
type	String	Rule types. The value can be: • BUILD_IN: built-in rules • USER_DEFINE: user-defined rules
regex	String	Regular expression
mask_value	String	Substitution value
status	String	Rule status
operate_time	String	Operation time

Status code: 400

Table 2-165 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-166 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-167 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-168 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-169 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-170 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{instance_id}/dbss/audit/sensitive/masks
```

Example Response

Status code: 200

Execution succeeded.

```
{
  "rules" : [ {
    "id" : "n34W2ngBo47GiyUSKOVl",
```



```
    "name" : "Passport NO.",
    "type" : "BUILD_IN",
    "regex" : "-",
    "mask_value" : "###",
    "status" : "ON",
    "operate_time" : "2030-01-01 00:00:06"
   }, {
    "id" : "nn4W2ngBo47GiyUSKOVP",
    "name" : "Military officer card NO.",
    "type" : "BUILD_IN",
    "regex" : "-",
    "mask_value" : "###",
    "status" : "ON",
    "operate_time" : "2030-01-01 00:00:05"
   }, {
    "id" : "nX4W2ngBo47GiyUSKOU9",
    "name" : "Ethnicity",
    "type" : "BUILD_IN",
    "regex" : "-",
    "mask_value" : "###",
    "status" : "ON",
    "operate_time" : "2030-01-01 00:00:04"
   }, {
    "id" : "mn4W2ngBo47GiyUSKOUO",
    "name" : "GPS Information",
    "type" : "BUILD_IN",
    "regex" : "-",
    "mask_value" : "###",
    "status" : "ON",
    "operate_time" : "2030-01-01 00:00:01"
   } ],
   "total" : 6
}
```


Status code: 400

Incorrect request parameter.


```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```


Status code: 500

Internal server error.


```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```


Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListAuditSensitiveMasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAuditSensitiveMasksRequest request = new ListAuditSensitiveMasksRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListAuditSensitiveMasksResponse response = client.listAuditSensitiveMasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
```

```
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAuditSensitiveMasksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_audit_sensitive_masks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAuditSensitiveMasksRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListAuditSensitiveMasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Incorrect request parameter.
403	Authentication failed.
500	Internal Server Error

Error Code

For details, see **Error Codes**.

2.6 TMS Tags

2.6.1 Querying Tags in a Project

Function

This API is used to query project tags.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{resource_type}/tags

Table 2-171 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_type	Yes	String	Resource type. • auditInstance

Request Parameter

Table 2-172 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 2-173 Response body parameter

Parameter	Parameter Type	Description
tags	Array of **tags** objects	Tags.

Table 2-174 tags

Parameter	Parameter Type	Description
key	String	Key. The value can contain up to 128 characters.
values	Array of strings	Specifies tag values. Each value can contain a maximum of 255 characters.

Status code: 400

Table 2-175 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-176 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-177 Response body parameter

Parameter	Response body parameter	Description
error	Object	Error message.

Table 2-178 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-179 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-180 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{resource_type}/tags
```

Example Response

Status code: 200

Execution succeeded.

```
{
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "value1", "value2" ]
  } ]
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;


public class ListProjectResourceTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
```



```
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListProjectResourceTagsRequest request = new ListProjectResourceTagsRequest();
        request.withResourceType("{resource_type}");
        try {
            ListProjectResourceTagsResponse response = client.listProjectResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListProjectResourceTagsRequest()
        request.resource_type = "{resource_type}"
        response = client.list_project_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
```



```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListProjectResourceTagsRequest{}
    request.ResourceType = "{resource_type}"
    response, err := client.ListProjectResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.6.2 Querying the Resource Instance List by Tag

Function

This API is used to query the resource instance list by tag.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{resource_type}/resource-instances/filter

Table 2-181 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_type	Yes	String	Resource type. • auditInstance

Table 2-182 Query parameter

Parameter	Mandatory	Parameter Type	Description
limit	No	String	Number of records to be queried, The maximum value of **limit** is **1000** and the minimum value is **1**.
offset	No	String	Index location. If **offset** is set to *N*, the query starts from the N+1 piece of data. The value is **0** by default, indicating that the query starts from the first piece of data. The value must be an integer and cannot be a negative number.

Request Parameter

Table 2-183 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-184 Request body parameter

Parameter	Mandatory	Parameter Type	Description
matches	No	Array of **matches** objects	Specifies the search criteria. The tag key is the parameter to match, for example, **resource_name**. The tag value indicates the value to be matched. The key is a fixed dictionary value and cannot contain duplicate keys or unsupported keys. Check whether fuzzy match is required based on the key value. For example, if key is set to resource_name, fuzzy search (case-insensitive) is performed by default. If value is empty, exact match is performed. Most services do not have resources without names. In this case, an empty list is returned. If **key** is **resource_id**, exact match is performed. Only **resource_name** for **key** is supported. Other **key** values will be available later.

Parameter	Mandatory	Parameter Type	Description
not_tags	No	Array of **TagKeyValuesBean** objects	The resources to be queried do not contain tags listed in not_tags. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns resources containing no tags in this list. Keys in this list are in an AND relationship while values in each key-value structure are in an OR relationship. If no tag filtering condition is specified, full data is returned.
tags	No	Array of **TagKeyValuesBean** objects	The resources to be queried contain tags listed in **tags**. Each instance to be queried contains a maximum of 50 keys. Each tag key has a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns instances containing all tags in this list. Keys in this list are in the AND relationship and values in each key-value structure are in the OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Parameter Type	Description
tags_any	No	Array of **TagKeyValuesBean** objects	The resources to be queried contain any tags listed in **tags_any**. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns instances containing any tag in this list. Keys in this list are in the OR relationship and values in each key-value structure are also in the OR relationship. If no tag filtering condition is specified, full data is returned.
not_tags_any	No	Array of **TagKeyValuesBean** objects	The resources to be queried do not contain any tags listed in not_tags_any. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns resources containing no tag in this list. Keys in this list are in the OR relationship and values in each key-value structure are also in the OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Parameter Type	Description
sys_tags	No	TagKeyValuesBean object	Only users with the **op_service** permission can use this parameter to filter resources. Only one tag structure is contained when this API is called by Tag Management Service (TMS). The key is **_sys_enterprise_project_idvalue**, and the value is an enterprise project ID. Currently, each key can contain only one value. **0** indicates the default enterprise project. **sys_tags** cannot be used together with tenant tag filtering criteria (**without_any_tag**, **tags**, **tags_any**, **not_tags**, and **not_tags_any**). If **sys_tags** is not specified, resources with all the tags specified in **tags** will be returned. If **tags** is not specified, all resources will be returned.
without_any_tag	No	Boolean	If this parameter is set to **true**, all resources without tags are queried. In this case, the **tags**, **tags_any**, **not_tags**, and **not_tags_any** fields are ignored.

Table 2-185 matches

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. Only resource_name is supported.
value	Yes	String	Value, which is the name of the resource to be matched.

Table 2-186 TagKeyValuesBean

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain a maximum of 128 Unicode characters. The tag key cannot be left blank. The system does not verify the character set of **key** when searching for resources. **key** cannot be empty, an empty string, or spaces. Before using **key**, delete single-byte character (SBC) spaces before and after the value.
values	Yes	Array of strings	List of tag values. Each value contains a maximum of 255 Unicode characters. Before verifying and using **values**, delete SBC spaces before and after the value. The value can be an empty array but cannot be left blank. If **values** is left blank, it indicates **any_value** (querying any value). The values are in the OR relationship.

Response Parameters

Status code: 200

Table 2-187 Response body parameter

Parameter	Type	Description
resources	Array of **resources** objects	Resource instance list.
total_count	Integer	Total number of records.

Table 2-188 resources

Parameter	Parameter Type	Description
resource_id	String	The disk ID.
resource_name	String	Resource name. If the resource does not have a name, the ID is returned.
resource_detail	Object	Resource details. This field is reserved for subsequent extension, and its value defaults to an empty string.
tags	Array of **tags** objects	A list of tags for queried resources to match against. This parameter is an empty array by default if there is no tag.
sys_tags	Array of **sys_tags** objects	Only the op_service permission can obtain this field. Currently, this field contains only the resource_tag structure. The key is _**sys_enterprise_project_id**. value: enterprise project ID. The value 0 indicates the default enterprise project. This parameter can only be used by users with the **op_service** permission.

Table 2-189 tags

Parameter	Parameter Type	Description
key	String	Tag key.
value	String	Value

Table 2-190 sys_tags

Parameter	Parameter Type	Description
key	String	Tag key.
value	String	Value

Status code: 400

Table 2-191 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-192 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-193 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-194 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-195 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-196 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{resource_type}/resource-instances/filter

{
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "sys_tags" : [ {
    "key" : "_sys_enterprise_project_id",
    "values" : [ "5aa119a8-d25b-45a7-8d1b-88e127885635" ]
  } ]
}
```

Example Responses

Status code: 200

Succeeded

```
{
  "resources" : [ {
    "resource_detail" : null,
    "resource_id" : "cdfs_cefs_wesas_12_dsad",
    "resource_name" : "resouece1",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    }, {
      "key" : "key2",
      "value" : "value1"
    } ],
    "sys_tags" : [ {
      "key" : "_sys_enterprise_project_id",
      "value" : "5aa119a8-d25b-45a7-8d1b-88e127885635"
    } ]
  } ],
  "total_count" : 1000
}
```


Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ListResourceInstanceByTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        ListResourceInstanceByTagRequest request = new ListResourceInstanceByTagRequest();
        request.withResourceType("{resource_type}");
        ResourceInstanceTagRequest body = new ResourceInstanceTagRequest();
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<TagKeyValuesBean> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<TagKeyValuesBean> listbodyTagsAny = new ArrayList<>();
```



```
        listbodyTagsAny.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value1");
        listTagsValues.add("value2");
        List<TagKeyValuesBean> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value1");
        listNotTagsValues.add("value2");
        List<TagKeyValuesBean> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<ResourceInstanceTagRequestMatches> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new ResourceInstanceTagRequestMatches()
                .withKey("resource_name")
                .withValue("resource1")
        );
        body.withSysTags("[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]");
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        request.withBody(body);
        try {
            ListResourceInstanceByTagResponse response = client.listResourceInstanceByTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListResourceInstanceByTagRequest()
        request.resource_type = "{resource_type}"
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value1",
            "value2"
        ]
        listTagsbody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            ResourceInstanceTagRequestMatches(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ResourceInstanceTagRequest(
            sys_tags="[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]",
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody
```



```
        )
        response = client.list_resource_instance_by_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourceInstanceByTagRequest{}
    request.ResourceType = "{resource_type}"
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
```



```
    var listTagsbody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    keyMatches:= "resource_name"
    valueMatches:= "resource1"
    var listMatchesbody = []model.ResourceInstanceTagRequestMatches{
        {
            Key: &keyMatches,
            Value: &valueMatches,
        },
    }
    var sysTagsSysTags interface{} = "[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]"
    request.Body = &model.ResourceInstanceTagRequest{
        SysTags: &sysTagsSysTags,
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
    }
    response, err := client.ListResourceInstanceByTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Success
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.6.3 Querying the Number of Resource Instances by Tag

Function

Querying the Number of Resource Instances by Tag

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{resource_type}/resource-instances/count

Table 2-197 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_type	Yes	String	Resource type. • auditInstance

Request Parameter

Table 2-198 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-199 Request body parameter

Parameter	Mandatory	Parameter Type	Description
matches	No	Array of **matches** objects	Specifies the search criteria. The tag key is the parameter to match, for example, **resource_name**. The tag value indicates the value to be matched. The key is a fixed dictionary value and cannot contain duplicate keys or unsupported keys. Check whether fuzzy match is required based on the key value. For example, if key is set to resource_name, fuzzy search (case-insensitive) is performed by default. If value is empty, exact match is performed. Most services do not have resources without names. In this case, an empty list is returned. If **key** is **resource_id**, exact match is performed. Only **resource_name** for **key** is supported. Other **key** values will be available later.
not_tags	No	Array of **TagKeyValuesBean** objects	The resources to be queried do not contain tags listed in not_tags. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns resources containing no tags in this list. Keys in this list are in an AND relationship while values in each key-value structure are in an OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Parameter Type	Description
tags	No	Array of **TagKeyValuesBean** objects	The resources to be queried contain tags listed in **tags**. Each instance to be queried contains a maximum of 50 keys. Each tag key has a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns instances containing all tags in this list. Keys in this list are in the AND relationship and values in each key-value structure are in the OR relationship. If no tag filtering condition is specified, full data is returned.
tags_any	No	Array of **TagKeyValuesBean** objects	The resources to be queried contain any tags listed in **tags_any**. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns instances containing any tag in this list. Keys in this list are in the OR relationship and values in each key-value structure are also in the OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Parameter Type	Description
not_tags_any	No	Array of **TagKeyValuesBean** objects	The resources to be queried do not contain any tags listed in not_tags_any. Each resource to be queried contains a maximum of 50 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Keys must be unique and values of a key must be unique. The response returns resources containing no tag in this list. Keys in this list are in the OR relationship and values in each key-value structure are also in the OR relationship. If no tag filtering condition is specified, full data is returned.
sys_tags	No	**TagKeyValuesBean** object	Only users with the **op_service** permission can use this parameter to filter resources. Only one tag structure is contained when this API is called by Tag Management Service (TMS). The key is **_sys_enterprise_project_idvalue**, and the value is an enterprise project ID. Currently, each key can contain only one value. **0** indicates the default enterprise project. **sys_tags** cannot be used together with tenant tag filtering criteria (**without_any_tag**, **tags**, **tags_any**, **not_tags**, and **not_tags_any**). If **sys_tags** is not specified, resources with all the tags specified in **tags** will be returned. If **tags** is not specified, all resources will be returned.

Parameter	Mandatory	Parameter Type	Description
without_any_tag	No	Boolean	If this parameter is set to **true**, all resources without tags are queried. In this case, the **tags**, **tags_any**, **not_tags**, and **not_tags_any** fields are ignored.

Table 2-200 matches

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. Only resource_name is supported.
value	Yes	String	Value, which is the name of the resource to be matched.

Table 2-201 TagKeyValuesBean

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain a maximum of 128 Unicode characters. The tag key cannot be left blank. The system does not verify the character set of **key** when searching for resources. **key** cannot be empty, an empty string, or spaces. Before using **key**, delete single-byte character (SBC) spaces before and after the value.

Parameter	Mandatory	Parameter Type	Description
values	Yes	Array of strings	List of tag values. Each value contains a maximum of 255 Unicode characters. Before verifying and using **values**, delete SBC spaces before and after the value. The value can be an empty array but cannot be left blank. If the values are null, it indicates **any_value** (querying any value). The values are in the OR relationship.

Response Parameters

Status code: 200

Table 2-202 Response body parameter

Parameter	Parameter Type	Description
resources	Array of **resources** objects	Resource instance list.
total_count	Integer	Total number of records.

Table 2-203 resources

Parameter	Parameter Type	Description
resource_id	String	Resource ID.
resource_name	String	Resource name. If the resource does not have a name, the ID is returned.
resource_detail	Object	Specifies the share details. This field is reserved for subsequent extension, and its value defaults to an empty string.
tags	Array of **tags** objects	A list of tags for queried resources to match against. This parameter is an empty array by default if there is no tag.

Parameter	Parameter Type	Description
sys_tags	Array of sys_tags objects	Only the op_service permission can obtain this field. Only one **resource_tag** structure is included currently. The key is **_sys_enterprise_project_id**. value: enterprise project ID. The value 0 indicates the default enterprise project. This field is not returned for the non-op_service permission.

Table 2-204 tags

Parameter	Parameter Type	Description
key	String	Tag key.
value	String	Value

Table 2-205 sys_tags

Parameter	Parameter Type	Description
key	String	Tag key.
value	String	Value

Status code: 400

Table 2-206 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-207 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-208 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-209 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-210 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-211 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{resource_type}/resource-instances/count

{
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "sys_tags" : [ {
    "key" : "_sys_enterprise_project_id",
    "values" : [ "5aa119a8-d25b-45a7-8d1b-88e127885635" ]
  } ]
}
```

Response Examples

Status code: 200

Execution succeeded.

```
{
  "total_count" : 1000
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
```



```
import com.huaweicloud.sdk.dbss.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CountResourceInstanceByTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        CountResourceInstanceByTagRequest request = new CountResourceInstanceByTagRequest();
        request.withResourceType("{resource_type}");
        ResourceInstanceTagRequest body = new ResourceInstanceTagRequest();
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<TagKeyValuesBean> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<TagKeyValuesBean> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value1");
        listTagsValues.add("value2");
        List<TagKeyValuesBean> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value1");
        listNotTagsValues.add("value2");
        List<TagKeyValuesBean> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new TagKeyValuesBean()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<ResourceInstanceTagRequestMatches> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new ResourceInstanceTagRequestMatches()
                .withKey("resource_name")
```



```
                .withValue("resource1")
        );
        body.withSysTags("[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]");
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        request.withBody(body);
        try {
            CountResourceInstanceByTagResponse response = client.countResourceInstanceByTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CountResourceInstanceByTagRequest()
        request.resource_type = "{resource_type}"
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
```

```
            "value2"
        ]
        listTagsAnybody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value1",
            "value2"
        ]
        listTagsbody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            TagKeyValuesBean(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            ResourceInstanceTagRequestMatches(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ResourceInstanceTagRequest(
            sys_tags="[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]",
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody
        )
        response = client.count_resource_instance_by_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```



```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CountResourceInstanceByTagRequest{}
    request.ResourceType = "{resource_type}"
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
    var listTagsbody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.TagKeyValuesBean{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    keyMatches:= "resource_name"
    valueMatches:= "resource1"
    var listMatchesbody = []model.ResourceInstanceTagRequestMatches{
        {
            Key: &keyMatches,
            Value: &valueMatches,
        },
    }
    var sysTagsSysTags interface{} = "[{\"values\":[\"5aa119a8-d25b-45a7-8d1b-88e127885635\"],\"key
\":\"_sys_enterprise_project_id\"}]"
    request.Body = &model.ResourceInstanceTagRequest{
```



```
        SysTags: &sysTagsSysTags,
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
    }
    response, err := client.CountResourceInstanceByTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.6.4 Adding Resource Tags in Batches

Function

This API is used to add resource tags in batches.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{resource_type}/{resource_id}/tags/create

Table 2-212 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_type	Yes	String	Resource type. • auditInstance
resource_id	Yes	String	Resource ID You can obtain the value from the resource_id field in the API for querying the instance list.

Request Parameter

Table 2-213 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token. The token can be queried by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 2-214 Request body parameter

Parameter	Mandatory	Parameter Type	Description
tags	No	Array of **tags** objects	Tags. This parameter is mandatory for the tenant permission. For the op_service permission, either tags or sys_tags can be set.

Parameter	Mandatory	Parameter Type	Description
sys_tags	No	Array of sys_tags objects	System tag list. This field is available only to the op_service permission. Choose either this field or tags. Currently, TMS invokes only one resource_tag structure. The key is fixed as _sys_enterprise_project_id. The value is **UUID** or **0**. **0** indicates the default enterprise project.

Table 2-215 tags

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain up to 128 characters.
value	Yes	String	Value. Each value can contain a maximum of 255 characters.

Table 2-216 sys_tags

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain up to 128 characters.
value	Yes	String	Value. Each value can contain a maximum of 255 characters.

Response Parameter

Status code: 204

Table 2-217 Response body parameters

Parameter	Parameter Type	Description
-	String	-

Status code: 400

Table 2-218 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-219 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-220 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-221 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-222 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-223 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{resource_type}/{resource_id}/tags/create

{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  } ]
}
```

Response Examples

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;

import java.util.List;
```



```
import java.util.ArrayList;

public class BatchAddResourceTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchAddResourceTagRequest request = new BatchAddResourceTagRequest();
        request.withResourceType("{resource_type}");
        request.withResourceId("{resource_id}");
        ResourceTagRequest body = new ResourceTagRequest();
        List<KeyValueBean> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new KeyValueBean()
                .withKey("key1")
                .withValue("value1")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchAddResourceTagResponse response = client.batchAddResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
```



```
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchAddResourceTagRequest()
        request.resource_type = "{resource_type}"
        request.resource_id = "{resource_id}"
        listTagsbody = [
            KeyValueBean(
                key="key1",
                value="value1"
            )
        ]
        request.body = ResourceTagRequest(
            tags=listTagsbody
        )
        response = client.batch_add_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchAddResourceTagRequest{}
    request.ResourceType = "{resource_type}"
    request.ResourceId = "{resource_id}"
    var listTagsbody = []model.KeyValueBean{
```



```
        {
            Key: "key1",
            Value: "value1",
        },
    }
    request.Body = &model.ResourceTagRequest{
        Tags: &listTagsbody,
    }
    response, err := client.BatchAddResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
204	Request succeeded.
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.6.5 Deleting Resource Tags in Batches

Function

This API is used to delete resource tags in batches.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v1/{project_id}/{resource_type}/{resource_id}/tags/delete

Table 2-224 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
resource_type	Yes	String	Resource type. • auditInstance
resource_id	Yes	String	Resource ID. You can obtain the value from the resource_id field in the API for querying the instance list.

Request Parameter

Table 2-225 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token.

Table 2-226 Request body parameter

Parameter	Mandatory	Parameter Type	Description
tags	No	Array of **tags** objects	Tags. This parameter is mandatory for the tenant permission. For the op_service permission, either tags or sys_tags can be set.

Parameter	Mandatory	Parameter Type	Description
sys_tags	No	Array of **sys_tags** objects	System tag list. This field is available only to the op_service permission. Choose either this field or tags. Currently, TMS invokes only one resource_tag structure. The key is fixed as _sys_enterprise_project_id. The value is **UUID** or **0**. **0** indicates the enterprise project by default. Currently, only **create** is supported.

Table 2-227 tags

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain up to 128 characters.
value	No	String	Value. Each value can contain a maximum of 255 characters.

Table 2-228 sys_tags

Parameter	Mandatory	Parameter Type	Description
key	Yes	String	Key. The value can contain up to 128 characters.
value	No	String	Value. Each value can contain a maximum of 255 characters.

Response Parameter

Status code: 204

Table 2-229 Response body parameter

Parameter	Parameter Type	Description
-	String	-

Status code: 400

Table 2-230 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-231 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-232 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-233 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-234 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-235 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example Request

```
/v1/{project_id}/{resource_type}/{resource_id}/tags/delete

{
  "tags" : [ {
    "key" : "key1"
  }, {
    "key" : "key2",
    "value" : "value3"
  } ]
}
```

Response Examples

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
import com.huaweicloud.sdk.dbss.v1.model.*;
```



```
import java.util.List;
import java.util.ArrayList;

public class BatchDeleteResourceTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchDeleteResourceTagRequest request = new BatchDeleteResourceTagRequest();
        request.withResourceType("{resource_type}");
        request.withResourceId("{resource_id}");
        ResourceTagRequest body = new ResourceTagRequest();
        List<KeyValueBean> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new KeyValueBean()
                .withKey("key1")
        );
        listbodyTags.add(
            new KeyValueBean()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchDeleteResourceTagResponse response = client.batchDeleteResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteResourceTagRequest()
        request.resource_type = "{resource_type}"
        request.resource_id = "{resource_id}"
        listTagsbody = [
            KeyValueBean(
                key="key1"
            ),
            KeyValueBean(
                key="key2",
                value="value3"
            )
        ]
        request.body = ResourceTagRequest(
            tags=listTagsbody
        )
        response = client.batch_delete_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dbss.NewDbssClient(
```



```
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteResourceTagRequest{}
    request.ResourceType = "{resource_type}"
    request.ResourceId = "{resource_id}"
    var listTagsbody = []model.KeyValueBean{
        {
            Key: "key1",
        },
        {
            Key: "key2",
            Value: "value3",
        },
    }
    request.Body = &model.ResourceTagRequest{
        Tags: &listTagsbody,
    }
    response, err := client.BatchDeleteResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
204	Request succeeded.
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

2.7 Adding an RDS Database (Deprecated)

Function

This API is used to add an RDS database.

NOTICE

This API is no longer maintained in V1 and will be brought offline. Use the API **Adding an RDS Database** of the V2 version.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{instance_id}/dbss/audit/databases/rds

Table 2-236 URI parameter

Parameter	Mandatory	Parameter Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID

Request Parameter

Table 2-237 Request header parameter

Parameter	Mandatory	Parameter Type	Description
X-Auth-Token	Yes	String	User token.

Table 2-238 Request body parameter

Parameter	Mandatory	Parameter Type	Description
databases	Yes	Array of **databases** objects	List of added databases
total_count	No	Integer	Total number

Table 2-239 databases

Parameter	Mandatory	Parameter Type	Description
id	Yes	String	Database ID.
db_name	Yes	String	Database name
status	Yes	String	Database status
port	Yes	String	Database port
ip	Yes	String	Database IP address
instance_name	Yes	String	DB instance name
version	Yes	String	Database version
type	Yes	String	DB type
enterprise_id	Yes	String	The enterprise project ID
enterprise_name	No	String	Enterprise project name

Response Parameters

Status code: 200

Table 2-240 Response body parameter

Parameter	Parameter Type	Description
illegal_db_id	Array of strings	ID of the database instance that fails to be added
legal_db_id	Array of strings	ID of the database instance that is successfully added

Status code: 400

Table 2-241 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-242 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 403

Table 2-243 Response body parameters

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-244 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Status code: 500

Table 2-245 Response body parameter

Parameter	Parameter Type	Description
error	Object	Error message.

Table 2-246 ErrorDetail

Parameter	Parameter Type	Description
error_code	String	Error code
error_msg	String	Error message

Example request

```
/v1/{project_id}/{instance_id}/dbss/audit/databases/rds

{
  "databases" : [ {
    "id" : "123751d3ee2f47aea64822e98318c6a8in01",
    "db_name" : "rds1",
    "status" : "ACTIVE",
    "port" : "3306",
    "ip" : "192.168.0.119",
    "instance_name" : "rds1",
    "version" : "8.0",
    "type" : "MySQL",
    "enterprise_id" : "0",
    "enterprise_name" : "default"
  }, {
    "id" : "2343f7285d684fed8b09fac201c3fc7ain01",
    "db_name" : "rds2",
    "status" : "ACTIVE",
    "port" : "3306",
    "ip" : "192.168.0.92",
    "instance_name" : "rds2",
    "version" : "8.0",
    "type" : "MySQL",
    "enterprise_id" : "0",
    "enterprise_name" : "default"
  } ]
}
```

Response Examples

Status code: 200

Execution succeeded.

```
{
  "illegal_db_id" : [ ],
  "legal_db_id" : [ "123751d3ee2f47aea64822e98318c6a8in01", "2343f7285d684fed8b09fac201c3fc7ain01" ]
}
```

Status code: 400

Failed

```
{
  "error" : {
    "error_code" : "DBSS.XXXX",
    "error_msg" : "XXX"
  }
}
```

Example SDK Code

The sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dbss.v1.region.DbssRegion;
import com.huaweicloud.sdk.dbss.v1.*;
```



```
import com.huaweicloud.sdk.dbss.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class AddRdsNoAgentDatabaseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        DbssClient client = DbssClient.newBuilder()
                .withCredential(auth)
                .withRegion(DbssRegion.valueOf("xx-xx"))
                .build();
        AddRdsNoAgentDatabaseRequest request = new AddRdsNoAgentDatabaseRequest();
        RdsNoAgentDbRequest body = new RdsNoAgentDbRequest();
        List<RdsNoAgentDbRequestDatabases> listbodyDatabases = new ArrayList<>();
        listbodyDatabases.add(
            new RdsNoAgentDbRequestDatabases()
                .withId("123751d3ee2f47aea64822e98318c6a8in01")
                .withDbName("rds1")
                .withStatus("ACTIVE")
                .withPort("3306")
                .withIp("192.168.0.119")
                .withInstanceName("rds1")
                .withVersion("8.0")
                .withType("MySQL")
                .withEnterpriseId("0")
                .withEnterpriseName("default")
        );
        listbodyDatabases.add(
            new RdsNoAgentDbRequestDatabases()
                .withId("2343f7285d684fed8b09fac201c3fc7ain01")
                .withDbName("rds2")
                .withStatus("ACTIVE")
                .withPort("3306")
                .withIp("192.168.0.92")
                .withInstanceName("rds2")
                .withVersion("8.0")
                .withType("MySQL")
                .withEnterpriseId("0")
                .withEnterpriseName("default")
        );
        body.withDatabases(listbodyDatabases);
        request.withBody(body);
        try {
            AddRdsNoAgentDatabaseResponse response = client.addRdsNoAgentDatabase(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdbss.v1.region.dbss_region import DbssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdbss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = DbssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DbssRegion.value_of("xx-xx")) \
        .build()

    try:
        request = AddRdsNoAgentDatabaseRequest()
        listDatabasesbody = [
            RdsNoAgentDbRequestDatabases(
                id="123751d3ee2f47aea64822e98318c6a8in01",
                db_name="rds1",
                status="ACTIVE",
                port="3306",
                ip="192.168.0.119",
                instance_name="rds1",
                version="8.0",
                type="MySQL",
                enterprise_id="0",
                enterprise_name="default"
            ),
            RdsNoAgentDbRequestDatabases(
                id="2343f7285d684fed8b09fac201c3fc7ain01",
                db_name="rds2",
                status="ACTIVE",
                port="3306",
                ip="192.168.0.92",
                instance_name="rds2",
                version="8.0",
                type="MySQL",
                enterprise_id="0",
                enterprise_name="default"
            )
        ]
        request.body = RdsNoAgentDbRequest(
            databases=listDatabasesbody
        )
        response = client.add_rds_no_agent_database(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dbss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dbss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := dbss.NewDbssClient(
        dbss.DbssClientBuilder().
            WithRegion(region.ValueOf("xx-xx")).
            WithCredential(auth).
            Build())

    request := &model.AddRdsNoAgentDatabaseRequest{}
    enterpriseNameDatabases:= "default"
    enterpriseNameDatabases1:= "default"
    var listDatabasesbody = []model.RdsNoAgentDbRequestDatabases{
        {
            Id: "123751d3ee2f47aea64822e98318c6a8in01",
            DbName: "rds1",
            Status: "ACTIVE",
            Port: "3306",
            Ip: "192.168.0.119",
            InstanceName: "rds1",
            Version: "8.0",
            Type: "MySQL",
            EnterpriseId: "0",
            EnterpriseName: &enterpriseNameDatabases,
        },
        {
            Id: "2343f7285d684fed8b09fac201c3fc7ain01",
            DbName: "rds2",
            Status: "ACTIVE",
            Port: "3306",
            Ip: "192.168.0.92",
            InstanceName: "rds2",
            Version: "8.0",
            Type: "MySQL",
            EnterpriseId: "0",
            EnterpriseName: &enterpriseNameDatabases1,
        },
    }
    request.Body = &model.RdsNoAgentDbRequest{
        Databases: listDatabasesbody,
    }
    response, err := client.AddRdsNoAgentDatabase(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```



```
  }
}
```

More

For more SDK code examples in various programming languages, see the **Sample Code** tab on the right of the **API Explorer** page, which can automatically generate the corresponding SDK code examples.

Status Code

Status Code	Description
200	Request succeeded.
400	Failed
403	Authentication failed.
500	Server error.

Error Codes

For details, see **Error Codes**.

3 Appendix

3.1 Status Codes

- Normal

Returned Value	Description
200	The request is successfully processed.

- Abnormal

Status Code	Error	Description
400	Bad Request	The server fails to process the request.
401	Unauthorized	A username and a password are required.
403	Forbidden	Access to the requested page is denied.
404	Not Found	The server fails to find the requested page.
405	Method Not Allowed	Method specified in the request is not allowed.
406	Not Acceptable	The response generated by the server could not be received by the client.
407	Proxy Authentication Required	Proxy authentication is required before the request is processed.
408	Request Timeout	The request timed out.
409	Conflict	The request is not processed due to a conflict.

Status Code	Error	Description
500	Internal Server Error	The request is not processed due to a server error.
501	Not Implemented	The request is not processed because the server does not support the requested function.
502	Bad Gateway	Failed to complete the request because the server has received an invalid response.
503	Service Unavailable	Failed to complete the request because the system is unavailable.
504	Gateway Timeout	A gateway timeout error occurs.

3.2 Error Code

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **Error Codes**.

Status Code	Error Code	Message	Description	Measure
400	DBSS.10000001	Enter a valid request message	The request is invalid.	Check parameters.
400	DBSS.10020101	Enter a valid request message	Failed to obtain the specification list.	Check parameters.
400	DBSS.10020102	Enter a valid request message	Perform operations on the database failed.	Check parameters.
400	DBSS.10020118	Failed to add database, exceeding the limit	Failed to add the database. The number of databases exceeds the upper limit.	Delete unnecessary databases or buy a new instance.
400	DBSS.10020140	Illegal order ID	The order ID does not meet requirements.	Check the order ID.

Status Code	Error Code	Message	Description	Measure
400	DBSS.10021016	Insufficient quota	Insufficient quota	Contact the administrator.
400	DBSS.10020021	Invalid request parameter ID.	Invalid request ID parameter.	Check parameters.
401	DBSS.10020100	Failed to authenticate the token in the request	Failed to authenticate the token carried in the request.	Check the token.
404	DBSS.10021004	ECS can not found the request page	The ECS server fails to find the requested page.	Check the ECS path configuration.
500	DBSS.11000000	Internal system exception. Contact technical support engineers	A system error occurs, please contact technical support engineers.	Contact the administrator.

3.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain the project ID by calling the API for **Querying Project Information Based on Specified Criteria**.

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. **{Endpoint}** is the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

In the following example, **id** indicates the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "xxxxxxxx",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
```



```
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```


Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following operations:

1. Log in to the management console.
2. Click the username and choose **My Credentials** from the drop-down list.
3. On the page, view the project ID in the project list.

Figure 3-1 Viewing project IDs