

Object Storage Service

Node.js SDK Developer Guide

Issue	01
Date	2025-05-19



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI 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.

Contents

1 Downloading an SDK (SDK for Node.js)	1
2 Example Programs (SDK for Node.js)	3
3 Getting Started (SDK for Node.js)	5
3.1 Before You Start (SDK for Node.js)	5
3.2 Setting Up an OBS Environment (SDK for Node.js)	5
3.3 Preparing a Development Environment (SDK for Node.js)	8
3.4 Installing the SDK (SDK for Node.js)	8
3.5 Obtaining Endpoints (SDK for Node.js)	9
3.6 Initializing an Instance of ObsClient (SDK for Node.js)	10
3.7 Creating a Bucket (SDK for Node.js)	11
3.8 Uploading an Object (SDK for Node.js)	11
3.9 Downloading an Object (SDK for Node.js)	12
3.10 Listing Objects (SDK for Node.js)	14
3.11 Deleting an Object (SDK for Node.js)	15
3.12 General Ways to Use ObsClient (SDK for Node.js)	16
4 Initialization (SDK for Node.js)	20
4.1 Configuring Access Keys (SDK for Node.js)	20
4.2 Creating an Instance of ObsClient (SDK for Node.js)	20
4.3 Configuring an Instance of ObsClient (SDK for Node.js)	23
4.4 Configuring SDK Logging (SDK for Node.js)	25
4.5 Configuring an SDK Proxy (SDK for Node.js)	25
5 Bucket Management (SDK for Node.js)	27
5.1 Creating a Bucket (SDK for Node.js)	27
5.2 Listing Buckets (SDK for Node.js)	39
5.3 Checking Whether a Bucket Exists (SDK for Node.js)	45
5.4 Deleting a Bucket (SDK for Node.js)	50
5.5 Obtaining Bucket Metadata (SDK for Node.js)	54
5.6 Configuring a Bucket ACL (SDK for Node.js)	62
5.7 Obtaining a Bucket ACL (SDK for Node.js)	75
5.8 Configuring a Bucket Policy (SDK for Node.js)	82
5.9 Obtaining the Policy of a Bucket (SDK for Node.js)	86
5.10 Deleting a Bucket Policy (SDK for Node.js)	91

5.11 Obtaining the Region of a Bucket (SDK for Node.js).....	95
5.12 Obtaining Storage Information of a Bucket (SDK for Node.js).....	99
5.13 Configuring a Storage Quota for a Bucket (SDK for Node.js).....	104
5.14 Obtaining the Storage Quota of a Bucket (SDK for Node.js).....	109
5.15 Configuring a Storage Class for a Bucket (SDK for Node.js).....	114
5.16 Obtaining the Storage Class of a Bucket (SDK for Node.js).....	119
6 Object Upload (SDK for Node.js).....	124
6.1 Object Upload Overview (SDK for Node.js).....	124
6.2 Uploading an Object - Text-Based (SDK for Node.js).....	125
6.3 Uploading an Object - Streaming (SDK for Node.js).....	142
6.4 Uploading an Object - File-Based (SDK for Node.js).....	161
6.5 Creating a Folder (SDK for Node.js).....	178
6.6 Configuring Object Metadata (SDK for Node.js).....	195
6.7 Configuring a Lifecycle Rule When Uploading an Object (SDK for Node.js).....	209
6.8 Uploading an Object - Append (SDK for Node.js).....	211
6.9 Uploading an Object - Resumable (SDK for Node.js).....	234
6.10 Uploading an Object - Browser-Based (SDK for Node.js).....	251
7 Object Download (SDK for Node.js).....	257
7.1 Overview (SDK for Node.js).....	257
7.2 Downloading an Object - Text-Based (SDK for Node.js).....	257
7.3 Downloading an Object - Streaming (SDK for Node.js).....	271
7.4 Downloading an Object - File-Based (SDK for Node.js).....	284
7.5 Downloading an Object - Range-Based (SDK for Node.js).....	297
7.6 Downloading an Object - Conditional (SDK for Node.js).....	310
7.7 Rewriting Response Headers (SDK for Node.js).....	323
7.8 Downloading an Archive Object (SDK for Node.js).....	325
7.9 Downloading an Object - Resumable (SDK for Node.js).....	331
7.10 Processing an Image (SDK for Node.js).....	344
8 Object Management (SDK for Node.js).....	358
8.1 Configuring Object Metadata (SDK for Node.js).....	358
8.2 Obtaining Object Metadata (SDK for Node.js).....	371
8.3 Configuring an Object ACL (SDK for Node.js).....	384
8.4 Obtaining the ACL of an Object (SDK for Node.js).....	398
8.5 Listing Objects in a Bucket (SDK for Node.js).....	407
8.6 Deleting an Object (SDK for Node.js).....	424
8.7 Batch Deleting Objects (SDK for Node.js).....	430
8.8 Copying an Object (SDK for Node.js).....	439
9 Multipart Upload (SDK for Node.js).....	464
9.1 Multipart Upload APIs (SDK for Node.js).....	464
9.2 Initiating a Multipart Upload (SDK for Node.js).....	467
9.3 Uploading a Part (SDK for Node.js).....	485

9.4 Assembling Parts (SDK for Node.js).....	495
9.5 Listing Uploaded Parts (SDK for Node.js).....	504
9.6 Listing Multipart Uploads (SDK for Node.js).....	514
9.7 Copying a Part (SDK for Node.js).....	526
9.8 Aborting a Multipart Upload (SDK for Node.js).....	534
10 Temporarily Authorized Access (SDK for Node.js).....	541
10.1 Using a Temporary URL for Authorized Access (SDK for Node.js).....	541
11 Versioning (SDK for Node.js).....	554
11.1 Versioning Overview (SDK for Node.js).....	554
11.2 Configuring Versioning for a Bucket (SDK for Node.js).....	554
11.3 Viewing the Versioning Status of a Bucket (SDK for Node.js).....	559
11.4 Obtaining an Object Version (SDK for Node.js).....	564
11.5 Copying an Object Version (SDK for Node.js).....	577
11.6 Restoring an Archive Object Version (SDK for Node.js).....	597
11.7 Listing Object Versions in a Bucket (SDK for Node.js).....	603
11.8 Setting an ACL for an Object Version (SDK for Node.js).....	623
11.9 Obtaining the ACL of an Object Version (SDK for Node.js).....	625
11.10 Deleting an Object Version (SDK for Node.js).....	626
12 Lifecycle (SDK for Node.js).....	629
12.1 Configuring Lifecycle Rules for a Bucket (SDK for Node.js).....	629
12.2 Obtaining the Lifecycle Rules of a Bucket (SDK for Node.js).....	644
12.3 Deleting the Lifecycle Rules of a Bucket (SDK for Node.js).....	651
13 Bucket CORS (SDK for Node.js).....	657
13.1 Configuring CORS for a Bucket (SDK for Node.js).....	657
13.2 Obtaining the CORS Configuration of a Bucket (SDK for Node.js).....	665
13.3 Deleting the CORS Configuration of a Bucket (SDK for Node.js).....	672
14 Logging (SDK for Node.js).....	677
14.1 Configuring Logging for a Bucket (SDK for Node.js).....	677
14.2 Obtaining the Logging Configuration of a Bucket (SDK for Node.js).....	689
15 Static Website Hosting (SDK for Node.js).....	697
15.1 Overview (SDK for Node.js).....	697
15.2 Configuring Static Website Hosting (SDK for Node.js).....	699
15.3 Obtaining Static Website Hosting Configurations (SDK for Node.js).....	710
15.4 Deleting Website Hosting Settings (SDK for Node.js).....	718
16 Tagging (SDK for Node.js).....	724
16.1 Setting Bucket Tags (SDK for Node.js).....	724
16.2 Obtaining Bucket Tags (SDK for Node.js).....	731
16.3 Deleting Bucket Tags (SDK for Node.js).....	737
17 Server-Side Encryption (SDK for Node.js).....	743

17.1 Overview (SDK for Node.js)..... 743

17.2 APIs (SDK for Node.js)..... 743

17.3 Code Examples (SDK for Node.js)..... 745

18 Troubleshooting (SDK for Node.js).....748

18.1 OBS Server-Side Error Codes (SDK for Node.js)..... 748

18.2 SDK Common Results (SDK for Node.js)..... 755

18.3 Log Analysis (SDK for Node.js)..... 757

18.4 Missing Modules (SDK for Node.js)..... 758

18.5 Connection Timeout (SDK for Node.js)..... 758

18.6 Unmatched Signatures (SDK for Node.js)..... 758

19 FAQs (SDK for Node.js)..... 759

19.1 How Do I Specify Content-SHA256? (SDK for Node.js)..... 759

19.2 Why Does the SDK Source Code Contain **acs.amazonaws.com**? (SDK for Node.js)..... 760

19.3 How Do I Get My Account ID and User ID? (SDK for Node.js)..... 760

19.4 What Is Content-Type (MIME)? (SDK for Node.js)..... 762

1 Downloading an SDK (SDK for Node.js)

Download

- Latest version: [Download](#)
- Earlier versions: [Download](#)

Compatibility

- Recommended versions: Node 0.12.x, Node4.x, Node6.x, Node8.x, or Node10.x
- Interface changes: The following table describes the interfaces not completely compatible with earlier versions 2.1.x:

Interface	Description
ObsClient.listBuckets	In the response, the data type of InterfaceResult.Buckets was changed to Array . InterfaceResult.Buckets.Bucket was deleted.
ObsClient.setBucketACL	In the request, the data type of Grants was changed to Array . Grants.Grant was deleted.
ObsClient.getBucketACL	In the response, the data type of InterfaceResult.Grants was changed to Array . InterfaceResult.Grants.Grant was deleted.
ObsClient.setObjectACL	In the request, the data type of Grants was changed to Array . Grants.Grant was deleted.
ObsClient.getObjectACL	In the response, the data type of InterfaceResult.Grants was changed to Array . InterfaceResult.Grants.Grant was deleted.
ObsClient.setBucketLogging	In the request, the data type of LoggingEnabled.TargetGrants was changed to Array . LoggingEnabled.TargetGrants.Grant was deleted.

Interface	Description
ObsClient.getBucketLogging	In the response, the data type of InterfaceResult.LoggingEnabled.TargetGrants was changed to Array . InterfaceResult.LoggingEnabled.TargetGrants.Grant was deleted.
ObsClient.setBucketWebsite	In the request, the data type of RoutingRules was changed to Array . RoutingRules.RoutingRule was deleted.
ObsClient.getBucketWebsite	In the response, the data type of InterfaceResult.RoutingRules was changed to Array . InterfaceResult.RoutingRules.RoutingRule was deleted.
ObsClient.setBucketCors	In the request, CorsRule was renamed as CorsRules .
ObsClient.getBucketCors	In the response, InterfaceResult.CorsRule was renamed as InterfaceResult.CorsRules .
ObsClient.setBucketTagging	In the request, the data type of TagSet was changed to Array . TagSet.Tag was deleted.
ObsClient.getBucketTagging	In the response, the data type of InterfaceResult.TagSet was changed to Array . InterfaceResult.TagSet.Tag was deleted.

2 Example Programs (SDK for Node.js)

OBS Node.js SDK provides abundant example programs for user reference and direct use. These programs can be obtained from the OBS Node.js SDK development package. For example, files in **eSDK_Storage_OBS_<VersionId>_Node.js/examples** obtained by decompressing **eSDK_Storage_OBS_<VersionId>_Node.js.zip** are example programs. Alternatively, you can click code package names provided in the following table to obtain corresponding example programs.

Example programs include:

Code Example	Description
bucket-operations-sample	How to use bucket-related APIs.
object-operations-sample	How to use object-related APIs.
download-sample	How to download an object.
create-folder-sample	How to create a folder.
delete-objects-sample	How to delete objects in a batch.
list-objects-sample	How to list objects.
list-versions-sample	How to list versioning objects.
list-objects-infolder-sample	How to list objects in a folder.
object-meta-sample	How to customize object metadata.
simple-multipart-upload-sample	How to perform a multipart upload.
restore-object-sample	How to download Archive objects.
concurrent-copypart-sample	How to concurrently copy parts of a large object.
concurrent-download-object-sample	How to concurrently download parts of a large object.

Code Example	Description
concurrent-uploadpart-sample	How to concurrently upload parts of a large object.
post-object-sample	How to perform a browser-based upload.
temporary-signature-sample	How to use URLs for authorized access.

3 Getting Started (SDK for Node.js)

3.1 Before You Start (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

- Ensure that you are familiar with OBS basic concepts, such as [buckets](#), [objects](#), and [access keys \(AK and SK\)](#).
- You can see [General Ways to Use ObsClient \(SDK for Node.js\)](#) to learn how to call OBS Node.js SDK APIs in a general manner.
- **ObsClient** supports API calling results returned via a callback function or the **Promise** object.
- Some features are available only in some regions. If an API call returns the 405 HTTP status code, check whether the region supports this feature.

3.2 Setting Up an OBS Environment (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Step 1 Sign up for a cloud service account.

Create an account to use OBS. If you already have one, use it instead.

1. Open a browser.
2. Visit the [Huawei Cloud official website](#).
3. In the upper right corner of the page, click **Register**.

4. Enter the registration information and click **Register**.

Step 2 Enable OBS.

Top up your account before you can use OBS.

1. Log in to the [management console](#).
2. Click **Billing & Costs** from the top menu bar. The **Billing Center** page is displayed.
3. Choose **Funds Management > Top Up**. The **Top Up** page is displayed.
4. Top up your account.
5. After the top-up is complete, close the dialog box and go back to the homepage.
6. Choose **Service List > Object Storage Service** to access OBS Console.

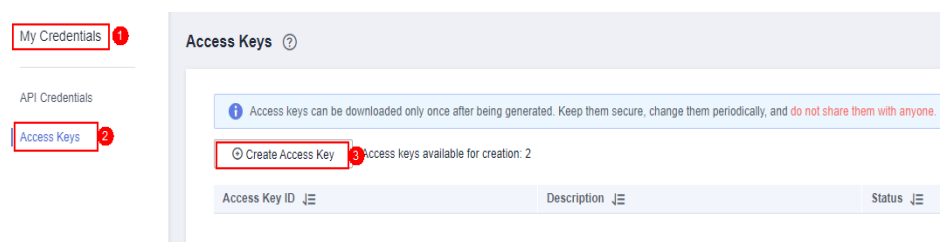
Step 3 Create access keys.

OBS employs access keys (AK and SK) for signature verification to ensure that only authorized accounts can access specified OBS resources. Detailed explanations of access keys are as follows:

- AK is short for Access Key ID. One AK maps to only one user but one user can have multiple AKs. OBS authenticates users by their AKs.
- SK is short for Secret Access Key, which is used to access OBS. You can generate authentication information based on SKs and request headers. An SK maps to an AK, and they group into a pair.

Access keys are permanent. There are also temporary security credentials (consisting of an AK/SK pair and a security token). Each user can create a maximum of two valid AK/SK pairs. Temporary security credentials can only be used to access OBS within the specified validity period. Once they expire, they must be requested again. For security purposes, you are advised to use temporary security credentials to access OBS. If you want to use permanent access keys, periodically update them.

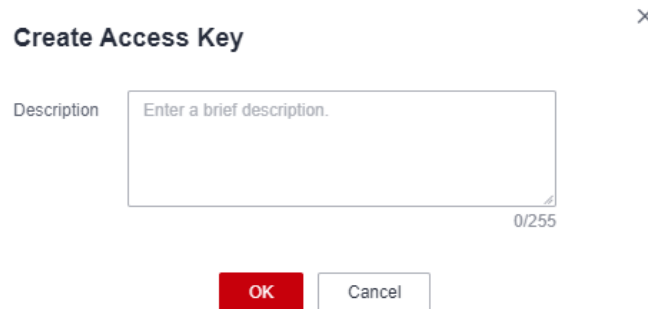
- To get permanent access keys, do as follows:
 - a. Log in to the [management console](#).
 - b. In the upper right corner, hover your cursor over the username and choose **My Credentials**.
 - c. On the **My Credentials** page, click **Access Keys** in the navigation pane.
 - d. On the **Access Keys** page, click **Create Access Key**.



NOTE

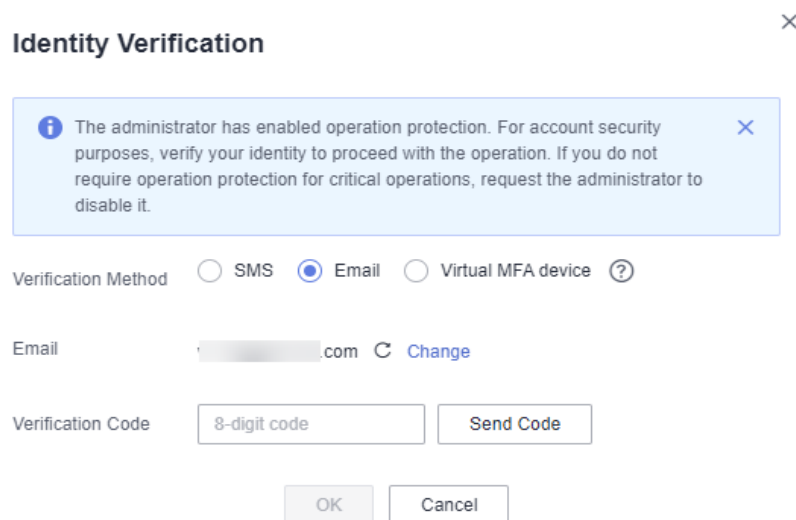
Each user can create a maximum of two valid AK/SK pairs.

- e. In the **Create Access Key** dialog box, enter a description (recommended), and click **OK**.



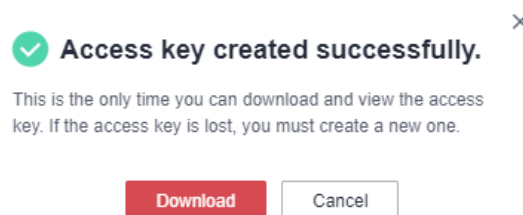
The **Create Access Key** dialog box features a title bar with a close button (X). Below the title is a label 'Description' followed by a text input field containing the placeholder text 'Enter a brief description.' and a character count '0/255' at the bottom right. At the bottom of the dialog are two buttons: a red 'OK' button and a white 'Cancel' button with a grey border.

- f. (Optional) In the displayed **Identity Verification** dialog box, select a verification method, enter the verification code, and click **OK**.



The **Identity Verification** dialog box has a title bar with a close button (X). Below the title is a blue information banner with a white 'i' icon, stating: 'The administrator has enabled operation protection. For account security purposes, verify your identity to proceed with the operation. If you do not require operation protection for critical operations, request the administrator to disable it.' Below the banner is the 'Verification Method' section with three radio buttons: 'SMS', 'Email' (which is selected), and 'Virtual MFA device' with a help icon (?). Below this is an 'Email' field showing a masked address and a 'Change' link. The 'Verification Code' section includes a text input field with the placeholder '8-digit code' and a 'Send Code' button. At the bottom are 'OK' and 'Cancel' buttons.

- g. In the displayed dialog box, click **Download** to save the access keys to your browser's default download path.



The dialog box shows a green checkmark icon and the text 'Access key created successfully.' Below this is a message: 'This is the only time you can download and view the access key. If the access key is lost, you must create a new one.' At the bottom are a red 'Download' button and a white 'Cancel' button with a grey border.

- h. Open the downloaded **credentials.csv** file to obtain the access keys (AK and SK).

 NOTE

- In the **credentials.csv** file, the AK is the value in the **Access Key ID** column, and the SK is the one in the **Secret Access Key** column.
- Keep the access keys properly to prevent information leakage. If you click **Cancel** in the download dialog box, the access keys will not be downloaded and cannot be downloaded later. You can create new access keys if required.

- To get temporary security credentials, refer to the following:
Temporary security credentials are issued by the system and are only valid for 15 minutes to 24 hours. They follow the principle of least privilege. When using temporary security credentials, you must use an AK/SK pair and a security token together.
To obtain them, see [Obtaining a Temporary AK/SK and a Security Token](#).

NOTICE

OBS is a global service. When obtaining temporary access keys, set the token scope to **domain** to apply the token to global services. Global services are not differentiated by any project or region.

-----End

3.3 Preparing a Development Environment (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

- Download the recommended version from the [Node.js's official website](#) and install it.
- Download the latest version of Eclipse IDE for JavaScript and Web Developer from [Eclipse's official website](#) and install it.

3.4 Installing the SDK (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

(Recommended) Installing the SDK Using npm

- Step 1** Run the **npm -V** command to check the npm version and ensure that the npm is installed.
- Step 2** Run the **npm install esdk-obs-nodejs** command to start the installation.

-----End

 NOTE

- On a Windows operating system, the message "Not internal or external command" is displayed when you run the npm command. In this case, add the npm installation directory (generally the installation directory of Node.js) to the Path environment variable.
- You may need to restart the computer for the environment variables to take effect.
- If you use npm to install dependencies and the network malfunctions, use proxies.

Installing the SDK with the Source Code

The following procedures use OBS Node.js SDK of the latest version as an example.

- Step 1** Download the OBS Node.js SDK by referring to [Downloading an SDK \(SDK for Node.js\)](#).
- Step 2** Decompress the development package to obtain folder **examples** (code examples), folder **lib** (SDK source code), file **package.json** (dependency configuration file), and file **README.txt** (feature description file of SDK versions).
- Step 3** On the command-line interface (CLI), go to the directory under which the SDK development package was decompressed, and run the **npm install** command to install dependency libraries. A **node_modules** folder will be generated.
- Step 4** (Optional) In the Eclipse JavaScript project, import the source code: Open Eclipse JavaScript IDE and choose **Import > Projects from Folder or Archive**. For **Import source**, select the directory under which the SDK development package is decompressed.

----End

 NOTE

After the installation, the directory structure is similar to the following:

```
|— examples
|— lib
|— node_modules
|— package.json
|— README.txt
```

3.5 Obtaining Endpoints (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

- You can click [here](#) to view the endpoints and regions enabled for OBS.

NOTICE

The SDK allows you to pass endpoints with or without the protocol name. Suppose the endpoint you obtained is **your-endpoint**. The endpoint passed when initializing an instance of **ObsClient** can be **http://your-endpoint**, **https://your-endpoint**, or **your-endpoint**.

3.6 Initializing an Instance of ObsClient (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Each time you want to send an HTTP/HTTPS request to OBS, you must create an instance of **ObsClient**. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Use the instance to access OBS.

// Close the ObsClient instance.
obsClient.close();
```

CAUTION

- JavaScript is an asynchronous programming language. Therefore, you cannot call the close method when accessing OBS.
- An ObsClient instance cannot be used again after it is closed by calling **obsClient.close**.

 NOTE

- For more information, see chapter "Initialization."
- For logging configuration, see [Configuring SDK Logging \(SDK for Node.js\)](#).
- For proxy configuration, see [Configuring an SDK Proxy \(SDK for Node.js\)](#).

3.7 Creating a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

A bucket is a global namespace of OBS and is a data container. It functions as a root directory of a file system and can store objects. The following code shows how to create a bucket:

 NOTE

- Bucket names are globally unique. Ensure that the bucket you create is named differently from any other bucket.
- A bucket name must comply with the following rules:
 - Contains 3 to 63 characters, chosen from lowercase letters, digits, hyphens (-), and periods (.), and starts with a digit or letter.
 - Cannot be an IP address.
 - Cannot start or end with a hyphen (-) or period (.).
 - Cannot contain two consecutive periods (..), for example, **my..bucket**.
 - Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
- If you create buckets of the same name, no error will be reported and the bucket properties comply with those set in the first creation request.
- For more information, see [Creating a Bucket \(SDK for Node.js\)](#).

3.8 Uploading an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

This example uploads string **Hello OBS** to bucket **examplebucket** as object **example/objectname**.

The example code is as follows:

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.
```

```
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify a text object.
      Body: 'Hello OBS'
    };
    // Upload the object.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

putObject();
```

NOTE

- For more information, see [Object Upload Overview \(SDK for Node.js\)](#).

3.9 Downloading an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

This example downloads object **example/objectname** from bucket **examplebucket**.

The example code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
    };

    // Download the object as text.
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("Object Content: %s", result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

getObject();
```

NOTE

- For more information, see [Overview \(SDK for Node.js\)](#).

3.10 Listing Objects (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

After objects are uploaded, you may want to view the objects contained in a bucket. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // List objects in a bucket.
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
```

```
listObjects();
```

NOTE

- In the sample code, 1000 objects will be listed, by default.
- For more information, see [Listing Objects](#).

3.11 Deleting an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

This example deletes object **example/objectname** from bucket **examplebucket**.

The example code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example) to delete.
      Key: 'example/objectname',
    };
    // Delete the object.
    const result = await obsClient.deleteObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
  } catch (error) {
    console.log(error);
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
deleteObject();
```

 NOTE

- For more information, see [Deleting an Object \(SDK for Node.js\)](#).

3.12 General Ways to Use ObsClient (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Result Returned via a Callback Function

ObsClient returns the results by using a callback function that contains two parameters in sequence: the exception information parameter and the [SDK common result object](#) parameter. If the exception information parameter in the callback function is not null, an error occurs during the API calling. Otherwise, the API is called. In such conditions, you need to obtain the HTTP status code from the [SDK common result object](#) parameter to check whether the operation is successful. Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Construct request parameters for bucket operations.
var requestParam1 = {
  Bucket: 'bucketname'
  // Other fields.
};
```

```
var callback1 = (err, result) => {
    // Process the result of a bucket-related API call.
};

// Call the APIs for bucket operations, such as creating a bucket.
obsClient.createBucket(requestParam1, callback1);

// Construct request parameters for object operations.
var requestParam2 = {
    Bucket : 'bucketname',
    Key : 'objectname'
    // Other fields.
};

var callback2 = (err, result) => {
    // Process the result of an object-related API call.
};

// Call an object-related API, such as the API for downloading an object.
obsClient.getObject(requestParam2, callback2);
```

NOTE

For APIs used for bucket operations, the **Bucket** parameter contained in the request object indicates the bucket name. For APIs used for object operations, the **Bucket** and **Key** parameters contained in the request object specify the bucket name and object name, respectively.

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
    // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
    // coding may result in leakage.
    // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
    // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
    // variables or import an AK/SK pair in other ways.
    // security_token: process.env.SECURITY_TOKEN,
    // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
    // as an example. Replace it with the one currently in use.
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Call APIs to perform operations, such as uploading an object.
obsClient.putObject({
    Bucket : 'bucketname',
    Key : 'objectname',
    Body : 'Hello OBS'
}, (err, result) => {
    // If the err parameter is not null, an error occurs during the API calling.
    if(err){
        console.log('Error-->' + err);
    }else{
        // If the exception information is null, the API call is complete. In such conditions, you need to
        // check the HTTP status code.
        if(result.CommonMsg.Status < 300){// The operation is successful.
            if(result.InterfaceResult){
                // Process the business logic after the operation is successful.
            }
        }
    }
});
```

```
    }else{// The operation fails. Obtain details about the exception.
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
      console.log('HostId-->' + result.CommonMsg.HostId);
      console.log('RequestId-->' + result.CommonMsg.RequestId);
    }
  };
});
```

Result Returned via the Promise Object

ObsClient supports results returned via the **Promise** object. If no exception is caught by the **catch** method of the **Promise** object, the API calling is complete. In such conditions, you need to obtain the HTTP status code from the **SDK Common Result Object** to check whether the operation is successful. The following is a code sample:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Construct request parameters for bucket operations.
var requestParam1 = {
  Bucket : 'bucketname'
  // Other fields.
};

// Call the APIs for bucket operations, such as creating a bucket.
var promise1 = obsClient.createBucket(requestParam1);
promise1.then((result) => {
  // Process the API call result.
}).catch((err)=>{
  // Rectify the fault.
});

// Construct request parameters for object operations.
var requestParam2 = {
  Bucket : 'bucketname',
  Key : 'objectname'
  // Other fields.
};

// Call an object-related API, such as the API for downloading an object.
var promise2 = obsClient.getObject(requestParam2);
promise2.then((result) => {
  // Process the API call result.
}).catch((err)=>{
```



```
// Rectify the fault.  
});
```

NOTE

For APIs used for bucket operations, the **Bucket** parameter contained in the request object indicates the bucket name. For APIs used for object operations, the **Bucket** and **Key** parameters contained in the request object specify the bucket name and object name, respectively.

Sample code:

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
  // coding may result in leakage.  
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
  // variables or import an AK/SK pair in other ways.  
  // security_token: process.env.SECURITY_TOKEN,  
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here  
  // as an example. Replace it with the one currently in use.  
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
// Call APIs to perform operations, such as uploading an object.  
obsClient.putObject({  
  Bucket : 'bucketname',  
  Key : 'objectname',  
  Body : 'Hello OBS'  
}).then((result) => {  
  // If no exception occurs and the API call is complete, check the HTTP status code.  
  if(result.CommonMsg.Status < 300){// Operation succeeded  
    if(result.InterfaceResult){  
      // Process the business logic after the operation is successful.  
    };  
  }  
} else{// The operation fails. Obtain details about the exception.  
  console.log('Code-->' + result.CommonMsg.Code);  
  console.log('Message-->' + result.CommonMsg.Message);  
  console.log('HostId-->' + result.CommonMsg.HostId);  
  console.log('RequestId-->' + result.CommonMsg.RequestId);  
};  
}).catch((err) => {  
  // An exception occurred after the API is called.  
  console.error('Error-->' + err);  
});
```

4 Initialization (SDK for Node.js)

4.1 Configuring Access Keys (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

To use OBS, you need a valid pair of AK and SK for signature authentication. For details, see [Setting Up an OBS Environment \(SDK for Node.js\)](#).

After obtaining the AK and SK, you can create an instance of `ObsClient` to call SDK APIs.

After obtaining the AK and SK, you can follow these steps to start initialization:

- [Creating an Instance of `ObsClient` \(SDK for Node.js\)](#)
- [Configuring an Instance of `ObsClient` \(SDK for Node.js\)](#)
- [Configuring SDK Logging \(SDK for Node.js\)](#)
- [Configuring an SDK Proxy \(SDK for Node.js\)](#)

4.2 Creating an Instance of `ObsClient` (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

ObsClient functions as the Node.js client for accessing OBS. It offers callers a series of APIs for interaction with OBS and is used for managing and operating resources, such as buckets and objects, stored in OBS. To use OBS Node.js SDK to send a request to OBS, you need to initialize an instance of **ObsClient** and modify parameters related to initial configurations of the instance based on actual needs.

By Using the Constructor

- Sample code for creating an ObsClient instance using permanent access keys (AKs/SKs):

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using
  // hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is
  // used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Use the instance to access OBS.

// Close the ObsClient instance.
// obsClient.close();
```

- Sample code for creating an ObsClient instance using temporary security credentials (AKs/SKs and security tokens):

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using
  // hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use
  // hard coding, which may result in information leakage. You can obtain an AK/SK pair using
  // environment variables or import an AK/SK pair in other ways.
  security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is
  // used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Use the instance to access OBS.

// Close the ObsClient instance.
// obsClient.close();
```

By Using the Factory Method

- Sample code for creating an ObsClient instance using permanent access keys (AKs/SKs):

```
// Import the OBS library.
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
```

```
// Initialize an ObsClient instance by using the factory method.
var obsClient = new ObsClient();
obsClient.factory({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using
  // hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is
  // used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Use the instance to access OBS.

// Close the ObsClient instance.
obsClient.close();
```

- Sample code for creating an ObsClient instance using temporary security credentials (AKs/SKs and security tokens):

```
// Import the OBS library.
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Initialize an ObsClient instance by using the factory method.
var obsClient = new ObsClient();
obsClient.factory({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using
  // hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use
  // hard coding, which may result in information leakage. You can obtain an AK/SK pair using
  // environment variables or import an AK/SK pair in other ways.
  security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is
  // used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

// Use the instance to access OBS.

// Close the ObsClient instance.
obsClient.close();
```

NOTE

- The project can contain one or more ObsClient instances.
- An ObsClient instance cannot be used again after it is closed by calling the close method.
- For details about SDK proxy configuration, see [Configuring an SDK Proxy \(SDK for Node.js\)](#).

4.3 Configuring an Instance of ObsClient (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

You can set the following initialization parameters to configure an instance of **ObsClient**.

Parameter	Description	Recommended Value
access_key_id	AK	N/A
secret_access_key	SK	N/A
server	Endpoint for accessing OBS, which contains the protocol type, domain name (or IP address), and port number. For example, https://your-endpoint:443. For security purposes, you are advised to use HTTPS.	N/A
max_retry_count	Maximum number of retries when an HTTP/HTTPS connection is abnormal. The default value is 3 .	[1, 5]
timeout	Timeout period (in seconds) of an HTTP/HTTPS request. The default value is 60 .	[10, 60]
http_agent	(Optional) Whether to use the HTTP agent. By default, this parameter is not configured.	N/A
https_agent	(Optional) Whether to use the HTTPS agent. By default, this parameter is not configured.	N/A

Parameter	Description	Recommended Value
ssl_verify	Whether to verify server-side certificates. Possible values are: • Path to the server-side root certificate file in .pem format • true: The default CAs are used to verify the server-side certificate. • false: The server-side certificates will not be verified. The default value is false .	N/A
long_conn_param	Persistent connection mode (in seconds). If the value is equal to or larger than 0 , the persistent connection mode is enabled and this value is used as the initial delay of the TCP Keep-Alive packets. By default, this parameter is left blank, which indicates that persistent connection mode is disabled.	N/A
is_cname	Whether to use self-defined domain name to access OBS. The default value is false .	N/A

 NOTE

- Parameters whose recommended value is **N/A** need to be set according to the actual conditions.
- If the network is unstable, you are advised to set a larger value for **timeout**.
- If the value of **server** does not contain any protocol, HTTPS is used by default.

NOTICE

- If the persistent connection mode is enabled, you must call **ObsClient.close** to close **ObsClient** explicitly to reclaim connection resources.
- For the sake of high DNS resolution performance and OBS reliability, you can set **server** only to the domain name of OBS, instead of the IP address.
- For details about SDK proxy configuration, see [Configuring an SDK Proxy \(SDK for Node.js\)](#).

4.4 Configuring SDK Logging (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

OBS Node.js SDK provides the logging function based on Log4js. You can call **ObsClient.initLog** to enable and configure logging. The following is a code sample:

```
obsClient.initLog({
  file_full_path: './logs/OBS-SDK.log', //Set the path to the log file.
  max_log_size: 20480, //Set the size of the log file, in bytes.
  backups: 10, //Set the maximum number of log files that can be stored.
  level: 'warn', //Set the log level.
  log_to_console: true //Set whether to print the log to console.
});
```

NOTE

- The logging function is disabled by default. You need to enable it manually.
- For details about SDK logs, see [Log Analysis](#).

4.5 Configuring an SDK Proxy (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Step 1 In the **root** directory of the project, run the following command to download the **proxy-agent** dependency.

```
npm install --save proxy-agent
```

Step 2 Declare and configure proxies in the project.

```
const ProxyAgent = require("proxy-agent");
const proxyAgent = new ProxyAgent(`http://username:password@proxyhost:proxyPort`);
```

Step 3 Initialize an ObsClient instance and configure a proxy.

```
var obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard
  // coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables
  // or import an AK/SK pair in other ways.
  security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
```

```
server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
https_agent: proxyAgent,  
http_agent: proxyAgent,  
});
```

----End

5 Bucket Management (SDK for Node.js)

5.1 Creating a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

OBS buckets are containers for storing objects you upload to OBS. This API creates a bucket.

When creating a bucket, you can also configure parameters such as the storage class, region, and access control list (ACL) as needed.

Restrictions

- To create a bucket, you must have the **obs:bucket:CreateBucket** permission. IAM is recommended for granting permissions. For details, see [IAM Custom Policies](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#). If the endpoint you use when creating a bucket indicates the default region CN North-Beijing1 (cn-north-1), you do not need to specify the region. Otherwise, you must specify the region based on the actual endpoint. To query region names, click [here](#). For example, if the endpoint you use for initialization is **obs.ap-southeast-1.myhuaweicloud.com**, the location you use when creating a bucket must be **ap-southeast-1**. Otherwise, status code 400 is returned.
- A maximum of 100 buckets (regardless of regions) can be created for an account. There is no limit on the number and size of objects in a bucket.
- A bucket name must be unique in OBS. If you repeatedly create buckets with the same name in the same region, an HTTP status code **200** will be returned. In other cases, creating a bucket with the same name as an existing bucket

will have an HTTP status code **409** returned, indicating that such a bucket already exists.

- The name of a deleted bucket can be reused for another bucket or a parallel file system at least 30 minutes after the deletion.
- Not all regions support the creation of multi-AZ buckets. You can check whether a region allows you to create multi-AZ buckets by referring to [Product Pricing Details](#).

Method

```
ObsClient.createBucket(params)
```

Request Parameters

Table 5-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Location	string	Yes if the OBS service region is not the default region	Explanation: Region where a bucket is located. Restrictions: If the used endpoint is obs.myhuaweicloud.com , this parameter is not required. If any other endpoint is used, this parameter is required. Value range: To learn about valid regions and endpoints, see Regions and Endpoints . An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator. Default value: If obs.myhuaweicloud.com is used as the endpoint and no region is specified, cn-north-1 (the CN North-Beijing1 region) is used by default.
ACL	AclType	No	Explanation: ACL that can be pre-defined when a bucket is created. For details about ACLs, see ACLs . Restrictions: None Value range: For details about the ACL options, see AclType . Default value: private
Storage Class	StorageClassType	No	Explanation: Bucket storage class that can be specified at bucket creation. Restrictions: None Value range: See StorageClassType . Default value: STANDARD

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can list objects, multipart uploads, and object versions in the bucket you are creating, as well as obtain bucket metadata. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js). Default value: None
GrantWrite	string	No	Explanation: ID (domain_id) of an account the WRITE permission is granted to. The account with the WRITE permission can create, delete, and overwrite objects in the bucket you are creating; initiate or abort multipart uploads; and upload, copy, and assemble parts. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js). Default value: None
GrantReadACP	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the bucket you are creating. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js). Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantWriteACP	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the bucket you are creating. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can perform any operation on the bucket you are creating. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
GrantReadDelivered	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. By default, this READ permission applies to all objects in the bucket. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantFullControlDelivered	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission has full control over the bucket you are creating. By default, the FULL_CONTROL permission applies to all objects in the bucket. Restrictions: None Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js). Default value: None
MultiEnterprise	string	No	Explanation: Enterprise project ID that can be specified during bucket creation. If you have enabled EPS, you can obtain the project ID from the EPS console. Restrictions: The value of epid is a UUID. epid is not required if you have not enabled EPS yet. Example: 9892d768-2d13-450f-aac7-ed0e44c2585f Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
AzRedundancy	string	No	Explanation: Data redundancy type that can be specified at bucket creation. Restrictions: Multi-AZ redundancy is not available for Archive storage. If the region where the bucket is located does not support multi-AZ storage, single-AZ storage is used by default. Value range: - To configure multi-AZ storage for the bucket, set this parameter to 3az. - To configure single-AZ storage (default value assigned by OBS) for the bucket, you do not need to specify this parameter. Default value: single AZ
FileInterface	string	No	Explanation: Whether a parallel file system is created Restrictions: None Value range: Enabled Default value: None. If this parameter is not specified, an object bucket is created.

Table 5-2 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 5-3 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 5-4 Responses

Type	Description
Table 5-5 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-5 .

Table 5-5 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-6 .

Parameter	Type	Description
InterfaceResult	BaseResponse Output	Explanation: Results outputted for a successful call. For details, see Table 5-7 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-6 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-7 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example creates a bucket named **examplebucket**. For bucket details, see the code comments below.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an ObsClient instance.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function createBucket() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the region where the bucket is to be created. The region must be the same as that in the
      // passed endpoint. ap-southeast-1 is used in this example.
      // Location: "ap-southeast-1",
      // Specify the bucket ACL. The following uses obs.AclPrivate as an example.
      ACL: obsClient.enums.AclPrivate,
      // Specify the storage class of the bucket. obs.StorageClassWarm (Infrequent Access) is used as an
      // example. If this parameter is not specified, the bucket is created with the Standard storage class.
      StorageClass: obsClient.enums.StorageClassWarm,
      // Specify the AZ type for the bucket. If this parameter is not specified, or it is specified but the involved
      // region does not support multi-AZ storage, single-AZ storage is applied. 3az is used in this example.
      AzRedundancy: "3az",
    };
    // Create a bucket.
    const result = await obsClient.createBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Create bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

createBucket();
```

5.2 Listing Buckets (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

OBS buckets are containers for storing objects you upload to OBS. This API returns a list of all buckets that meet the specified conditions in all regions of the current account. Returned buckets are listed in alphabetical order by bucket name.

Restrictions

- To list buckets, you must have the **obs:bucket:ListAllMyBuckets** permission. IAM is recommended for granting permissions. For details, see [IAM Custom Policies](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.listBuckets(params)
```

Request Parameters

Table 5-8 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
QueryLocation	boolean	No	Explanation: Whether to query the bucket location Restrictions: None Value range: true: The bucket location is queried. false: The bucket location is not queried. Default value: false

Parameter	Type	Mandatory (Yes/No)	Description
BucketType	string	No	Explanation: Type of bucket. Restrictions: None Value range: See BucketType . Default value: None. If this parameter is not specified, all buckets and parallel file systems are listed.

Table 5-9 BucketType

Constant	Description
OBJECT	An object bucket
POSIX	A parallel file system (POSIX)

Responses

Table 5-10 Responses

Type	Description
Table 5-11 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-11 .

Table 5-11 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-12 .

Parameter	Type	Description
InterfaceResult	ListBucketsOutput	Explanation: Results outputted for a successful call. For details, see Table 5-13 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-12 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-13 ListBucketsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server Default value: None
Owner	Owner	Explanation: Bucket owner Value range: See Table 5-14 .
Buckets	Bucket[]	Explanation: Bucket information list Value range: See Table 5-15 .

Table 5-14 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 5-15 Bucket

Parameter	Type	Description
BucketName	string	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. Default value: None
CreationDate	string	Explanation: When the bucket was created. Default value: None

Parameter	Type	Description
Location	string	Explanation: Where a bucket is located. Restrictions: If the used endpoint is obs.myhuaweicloud.com, this parameter is not required. If any other endpoint is used, this parameter is required. Value range: To learn about valid regions and endpoints, see Regions and Endpoints. An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator. Default value: If obs.myhuaweicloud.com is used as the endpoint and no region is specified, cn-north-1 (the CN North-Beijing1 region) is used by default.
BucketType	string	Explanation: Bucket type Value range: - Object bucket: OBJECT - Parallel file system: POSIX Default value: None

Code Examples

You can call **ObsClient.listBuckets** to list buckets. This example lists all buckets.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
```

```
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listBuckets() {
  try {
    const params = {
      // Specify whether to query the bucket location in the listing. The default value is false. true is used in
      // this example.
      QueryLocation: true,
      // Specify a bucket type (object buckets in this example). If this parameter is not specified, object
      // buckets and parallel file systems are listed by default.
      BucketType: "OBJECT",
    };
    // List buckets.
    const result = await obsClient.listBuckets(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List buckets successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Owner:');
      console.log('ID: %s', result.InterfaceResult.Owner.ID);
      console.log('Name: %s', result.InterfaceResult.Owner.Name);
      console.log('Buckets:');
      for (let i = 0; i < result.InterfaceResult.Buckets.length; i++) {
        const val = result.InterfaceResult.Buckets[i];
        console.log("Bucket[%d]-Name:%s,CreationDate:%s,Location: %s",
          i, val.BucketName, val.CreationDate, val.Location);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

listBuckets();
```

5.3 Checking Whether a Bucket Exists (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API checks whether a bucket exists. If an HTTP status code **200** is returned, the bucket exists. If **404** is returned, the bucket does not exist.

Restrictions

- To check whether a bucket exists, you must be the bucket owner or have the required permission (**obs:bucket:HeadBucket** in IAM or **HeadBucket** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.headBucket(params)
```

Request Parameters

Table 5-16 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-17 Responses

Type	Description
Table 5-18 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-18 .

Table 5-18 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-19 .
InterfaceResult	BaseResponse Output	Explanation: Results outputted for a successful call. For details, see Table 5-20 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-19 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-20 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.headBucket** to check for a bucket. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function headBucket() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Check whether the bucket exists.
    const result = await obsClient.headBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Head bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
headBucket();
```

5.4 Deleting a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API deletes an empty bucket. You can delete buckets you no longer use to free up space. The name of a deleted bucket can be reused for another bucket at least 30 minutes after the deletion.

NOTE

The data you proactively deleted from OBS cannot be recovered. Exercise caution when using this API.

Restrictions

- Only empty buckets can be deleted. An empty bucket means that:
 - The bucket does not contain any objects (including historical versions and delete markers).
 - The bucket does not contain any fragments, which means that there are no multipart uploads that have not been completed in the bucket.
- Bucket deletion is a non-idempotence operation and will fail if the to-be-deleted bucket does not exist.
- To delete a bucket, you must be the bucket owner or have the required permission (**obs:bucket:DeleteBucket** in IAM or **DeleteBucket** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.deleteBucket(params)
```


Request Parameters

Table 5-21 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-22 Responses

Type	Description
Table 5-23 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-23 .

Table 5-23 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-24 .
InterfaceResult	BaseResponse Output	Explanation: Results outputted for a successful call. For details, see Table 5-25 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-24 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-25 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.deleteBucket** to delete a bucket. This example deletes bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucket() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Delete a bucket.
    const result = await obsClient.deleteBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
deleteBucket();
```

5.5 Obtaining Bucket Metadata (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API returns information about a bucket, including the storage class, region, CORS rules, and redundancy policy.

Restrictions

- To obtain bucket metadata, you must be the bucket owner or have the required permission (**obs:bucket:HeadBucket** in IAM or **HeadBucket** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

ObsClient.getBucketMetadata(params)

Request Parameters

Table 5-26 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-27 Responses

Type	Description
Table 5-28 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-28 .

Table 5-28 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-29 .
InterfaceResult	GetBucketMetadataOutput	Explanation: Results outputted for a successful call. For details, see Table 5-30 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-29 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-30 GetBucketMetadataOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server Default value: None
StorageClass	StorageClassType	Explanation: Storage class of the bucket. Value range: See Table 5-31 . Default value: If the storage class of the bucket is Standard, this parameter is not specified.
Location	string	Explanation: Region where a bucket is located. Restrictions: If the used endpoint is obs.myhuaweicloud.com , this parameter is not required. If any other endpoint is used, this parameter is required. Value range: To learn about valid regions and endpoints, see Regions and Endpoints . An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator. Default value: If obs.myhuaweicloud.com is used as the endpoint and no region is specified, cn-north-1 (the CN North-Beijing1 region) is used by default.

Parameter	Type	Description
ObsVersion	string	Explanation: OBS version of the bucket. Value range: • 3.0 indicates the latest OBS version. • -- indicates any version earlier than 3.0. Default value: None
AllowOrigin	string	Explanation: If Origin in the request meets the CORS rules of the bucket, AllowedOrigin specified in the CORS rules is returned. AllowedOrigin indicates the origin from which requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain at most one wildcard character (*). Example: https://*.vbs.example.com Default value: None
AllowHeader	string	Explanation: If RequestHeader in the request meets the CORS rules of the bucket, AllowedHeader specified in the CORS rules is returned. AllowedHeader indicates the allowed headers for cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Default value: None

Parameter	Type	Description
AllowMethod	string	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method allowed for cross-origin requests, that is, the operation type of buckets and objects. Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None
ExposeHeader	string	Explanation: ExposeHeader in the CORS rules of the bucket. It specifies the CORS-allowed additional headers in the response. These headers provide additional information to clients. By default, your browser can only access headers Content-Length and Content-Type. If your browser needs to access other headers, add them to a list of the allowed additional headers. Restrictions: Spaces, asterisks (*), ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Default value: None
MaxAgeSeconds	number	Explanation: MaxAgeSeconds in the CORS rules of the bucket. It specifies the time your client can cache the response for a cross-origin request. Restrictions: Each CORS rule can contain at most one MaxAgeSeconds. Value range: 0 to ($2^{31} - 1$), in seconds Default value: 100

Parameter	Type	Description
MultiEnterprise	string	Explanation: Enterprise project ID that can be specified during bucket creation. If you have enabled EPS, you can obtain the project ID from the EPS console. Restrictions: The value of epid is a UUID. epid is not required if you have not enabled EPS yet. Example: 9892d768-2d13-450f-aac7-ed0e44c2585f Value range: To obtain the enterprise project ID, see How Do I Obtain an Enterprise Project ID? Default value: None
AzRedundancy	string	No Explanation: Data redundancy type that can be specified at bucket creation. Restrictions: Multi-AZ redundancy is not available for Archive storage. If the region where the bucket is located does not support multi-AZ storage, the bucket adopts single-AZ storage by default. Value range: To configure multi-AZ storage for the bucket, set this parameter to 3az. To configure single-AZ storage (default value assigned by OBS) for the bucket, you do not need to specify this parameter. Default value: single AZ

Table 5-31 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

You can call **ObsClient.getBucketMetadata** to obtain the metadata of a bucket. This example uses bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketMetadata() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Obtain the bucket metadata.
    const result = await obsClient.getBucketMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s metadata successful!", params.Bucket);
    }
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
console.log('StorageClass: %s', result.InterfaceResult.StorageClass);
console.log('Location: %s', result.InterfaceResult.Location);
console.log('AZRedundancy: %s', result.InterfaceResult.AZRedundancy);
console.log('Epid: %s', result.InterfaceResult.Epid);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketMetadata();
```

5.6 Configuring a Bucket ACL (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Access control lists (ACLs) allow resource owners to grant other accounts the permissions to access resources. By default, only the resource owner has full control over resources when a bucket or object is created. That is, the bucket creator has full control over the bucket, and the object uploader has full control over the object. Other accounts do not have the permissions to access resources. If resource owners want to grant other accounts the read and write permissions on resources, they can use ACLs. ACLs grant permissions to accounts. After an account is granted permissions, both the account and its IAM users can access the resources.

For more information, see [ACLs](#).

This API modifies a bucket ACL.

Restrictions

- A bucket can have up to 100 ACL rules.
- To configure an ACL for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketAcl** in IAM or **PutBucketAcl** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketAcl(params)
```

Request Parameters

Table 5-32 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ACL	AclType	No	Explanation: Pre-defined ACL. Restrictions: You must specify either ACL or the combination of Owner and Grants. Value range: See Table 5-33. Default value: None
Owner	Owner	No	Explanation: Bucket owner Restrictions: • Owner and Grants must be used together, and they cannot be used with ACL. • You must specify either ACL or the combination of Owner and Grants. Value range: See Table 5-34. Default value: None
Grants	Grant []	No	Explanation: Grantees and permissions Restrictions: • Owner and Grants must be used together, and they cannot be used with ACL. • You must specify either ACL or the combination of Owner and Grants. Value range: See Table 5-35. Default value: None

Table 5-33 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 5-34 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js). Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 5-35 Grant

Parameter	Type	Mandatory (Yes/No)	Description
Grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 5-36 .
Permission	PermissionType	Yes if used as a request parameter	Explanation: Granted permission Value range: See Table 5-39 . Default value: None
Delivered	boolean	No	Explanation: Whether the ACL of the bucket applies to its objects Value range: • true: The ACL of the bucket applies to its objects. • false: The ACL of the bucket does not apply to its objects. Default value: None

Table 5-36 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
Type	string	Yes if used as a request parameter	Explanation: Grantee type Value range: See Table 5-37 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if this parameter is used as a request parameter and Type is set to a user	Explanation: Account (domain) ID of the grantee. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Name	string	No if used as a request parameter	Explanation: Account name of the grantee Restrictions: • The account name cannot contain full-width characters. • The account name starts with a letter. • The account name contains 6 to 32 characters. • The account name contains only letters, digits, hyphens (-), or underscores (_). Default value: None
URI	Group UriType	Yes if this parameter is used as a request parameter and Type is set to a group	Explanation: Authorized user group Value range: See Table 5-38 . Default value: None

Table 5-37 GranteeType

Constant	Description
Group	Grants permissions to user groups.
CanonicalUser	Grants permissions to individual users.

Table 5-38 GroupUriType

Constant	Default Value	Description
ObsClient.enums.GroupAllUsers	AllUsers	All users.
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.
ObsClient.enums.GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 5-39 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadAcp	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.

Constant	Default Value	Description
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Responses

Table 5-40 Responses

Type	Description
Table 5-41 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-41 .

Table 5-41 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-42 .
InterfaceResult	Table 5-43	Explanation: Results outputted for a successful call. For details, see Table 5-43 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-42 ICommonMsg

Parameter	Type	Description

Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-43 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples: Setting a Pre-defined ACL When Creating a Bucket

This example sets a pre-defined ACL (private read and write) when creating a bucket.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an ObsClient instance.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
```

```
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function createBucket() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the region where the bucket is to be created. The region must be the same as that in the
      // passed endpoint. ap-southeast-1 is used in this example.
      // Location: "ap-southeast-1",
      // Specify the ACL. obs.AclPrivate is used as an example.
      ACL: obsClient.enums.AclPrivate,
    };
    // Create a bucket.
    const result = await obsClient.createBucket(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Create bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

createBucket();
```

Code Examples: Setting a Pre-defined ACL for a Bucket

This example sets an ACL (private) for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  //coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
async function setBucketAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Set the bucket ACL to be private.
      ACL: obsClient.enums.AclPrivate
    };
    // Set the bucket ACL.
    const result = await obsClient.setBucketAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s acl successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketAcl();
```

Code Examples: Setting an ACL for a Bucket Directly

This example sets an ACL to allow all users to read from bucket **examplebucket** but only allow user **0a03f5833900d3730f13c00f49d5exxx** to write to the bucket.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the bucket owner ID. ownerid is used in this example.
      Owner: { ID: 'ownerid' },
      Grants: [
        // Grant the public-read permission to all users.
        { Grantee: { Type: 'Group', URI: obsClient.enums.GroupAllUsers }, Permission:
```



```
obsClient.enums.PermissionRead },
    // Grant the write permission to a specific user. In this example, the user ID is
    0a03f5833900d3730f13c00f49d5exxx.
    { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
obsClient.enums.PermissionWrite },
    ]
};
// Set the bucket ACL.
const result = await obsClient.setBucketAcl(params);
if (result.CommonMsg.Status <= 300) {
    console.log("Set bucket(%s)'s acl successful!", params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    return;
};

console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
setBucketAcl();
```

5.7 Obtaining a Bucket ACL (SDK for Node.js)

Function

Access control lists (ACLs) allow resource owners to grant other accounts the permissions to access resources. By default, only the resource owner has full control over resources when a bucket or object is created. That is, the bucket creator has full control over the bucket, and the object uploader has full control over the object. Other accounts do not have the permissions to access resources. If resource owners want to grant other accounts the read and write permissions on resources, they can use ACLs. ACLs grant permissions to accounts. After an account is granted permissions, both the account and its IAM users can access the resources.

For more information, see [ACLs](#).

This API returns the ACL of a bucket.

Restrictions

- To obtain the ACL of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketAcl** in IAM or **GetBucketAcl** in a bucket policy). For details, see [Introduction to OBS Access Control, IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketAcl(params)
```

Request Parameters

Table 5-44 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-45 Responses

Type	Description
Table 5-46 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-46 .

Table 5-46 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-47 .
InterfaceResult	Table 5-48	Explanation: Results outputted for a successful call. For details, see Table 5-48 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-47 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-48 GetBucketAclOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Owner	Owner	Explanation: Account ID of the bucket owner. For details, see Table 5-49 .
Grants	Grant[]	Explanation: Grantees' permission information. For details, see Table 5-50 .

Table 5-49 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 5-50 Grant

Parameter	Type	Description
Grantee	Grantee	Explanation: Grantee information. For details, see Table 5-51 .
Permission	string	Explanation: Granted permissions. For details, see Table 5-54 .

Table 5-51 Grantee

Parameter	Type	Description
Type	string	Explanation: Grantee type. For details, see Table 5-52 .
ID	string	Explanation: Account (domain) ID of the grantee.
Name	string	Explanation: Account name of the grantee
URI	string	Explanation: Authorized user group. For details, see Table 5-53 .

Table 5-52 GranteeType

Constant	Description
Group	Grants permissions to user groups.
CanonicalUser	Grants permissions to individual users.

Table 5-53 GroupUriType

Constant	Default Value	Description
ObsClient.enums.GroupAllUsers	AllUsers	All users.
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.

Constant	Default Value	Description
ObsClient.enums.GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 5-54 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadAcp	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Code Examples

This example returns the ACL of bucket **examplebucket**.

```
// Import the OBS library.  
// Use npm to install the client.
```

```
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Obtain the bucket ACL.
    const result = await obsClient.getBucketAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s acl rules successful!", params.Bucket);
      console.log("RequestId: %s", result.InterfaceResult.RequestId);
      console.log("Owner[ID]: %s", result.InterfaceResult.Owner.ID);
      console.log("Grants:");
      for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
        const grant = result.InterfaceResult.Grants[i];
        console.log("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s",
          i, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission
        );
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

getBucketAcl();
```

Helpful Links

- [Obtaining the Bucket ACL](#)
- [OBS Error Codes](#)
- [Access Control FAQ](#)

5.8 Configuring a Bucket Policy (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

OBS provides access control over buckets. You can use an access policy to define whether a user can perform certain operations on a specific bucket. OBS access control can be implemented using IAM permissions, bucket policies, and ACLs. For more information, see [Introduction to OBS Access Control](#).

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permission for the bucket and the objects in it to IAM users or other accounts. If you want IAM users to have different permissions for different buckets, you need to configure different bucket policies for those users.

This API configures a policy for a bucket.

Restrictions

- Permissions for creating a bucket and obtaining a bucket list are service level and should be granted using [IAM Permissions](#).
- Due to data caching, after a bucket policy is configured, it takes up to five minutes for the policy to take effect.
- To configure a bucket policy, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketPolicy** in IAM or **PutBucketPolicy** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketPolicy(params)
```


Request Parameters

Table 5-55 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Policy	string	Yes	Explanation: Policy in JSON format Restrictions: - The bucket name contained in the Resource parameter of the policy must be the same as the one specified for the current bucket policy. - For details about the policy format, see Bucket Policy Parameters. Value range: See Bucket Policy Parameters . Default value: None

Responses

Table 5-56 Responses

Type	Description
Table 5-57 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-57 .

Table 5-57 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-58 .

Parameter	Type	Description
InterfaceResult	Table 5-59	Explanation: Results outputted for a successful call. For details, see Table 5-59 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-58 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-59 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples: Specifying a Bucket Policy

You can call **ObsClient.setBucketPolicy** to set a bucket policy. Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketPolicy() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify a bucket policy.
      Policy: "{\"Statement\":[{\"Sid\":\"Custom-policy-2482\",\"Effect\":\"Allow\",\"Principal\":{\"ID\":\"*\"}},{\"Action\":\"*\",\"ListBucket\"}],\"Resource\":[\"examplebucket\"]}",
    };
    // Set a bucket policy.
    const result = await obsClient.setBucketPolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s policy successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

setBucketPolicy();
```

5.9 Obtaining the Policy of a Bucket (SDK for Node.js)

Function

OBS provides access control over buckets. You can use an access policy to define whether a user can perform certain operations on a specific bucket. OBS access

control can be implemented using IAM permissions, bucket policies, and ACLs. For more information, see [Introduction to OBS Access Control](#).

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permission for the bucket and the objects in it to IAM users or other accounts. If you want IAM users to have different permissions for different buckets, you need to configure different bucket policies for those users.

This API returns the policy of a bucket.

Restrictions

- OBS returns "404 NoSuchBucketPolicy" when you call this API in the following scenarios:
 - The specified bucket policy does not exist.
 - The standard policy of the specified bucket is set to **Private** and no advanced policies are configured.
- To obtain the policy of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketPolicy** in IAM or **GetBucketPolicy** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketPolicy(params)
```

Request Parameters

Table 5-60 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-61 Responses

Type	Description
Table 5-62 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-62 .

Table 5-62 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-63 .
InterfaceResult	Table 5-64	Explanation: Results outputted for a successful call. For details, see Table 5-64 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-63 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-64 GetBucketPolicyOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Policy	string	Explanation: Policy in JSON format Restrictions: - The bucket name contained in the Resource parameter of the policy must be the same as the one specified for the current bucket policy. - For details about the policy format, see Bucket Policy Parameters.

Code Examples

This example returns the policy of bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketPolicy() {
```



```
try {
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket"
  };
  // Obtain the bucket policy.
  const result = await obsClient.getBucketPolicy(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Get bucket(%s)'s policy successful!", params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log("Policy: %s", result.InterfaceResult.Policy);
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketPolicy();
```

5.10 Deleting a Bucket Policy (SDK for Node.js)

Function

OBS provides access control over buckets. You can use an access policy to define whether a user can perform certain operations on a specific bucket. OBS access control can be implemented using IAM permissions, bucket policies, and ACLs. For more information, see [Introduction to OBS Access Control](#).

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permission for the bucket and the objects in it to IAM users or other accounts. If you want IAM users to have different permissions for different buckets, you need to configure different bucket policies for those users.

This API deletes the policy of a bucket. OBS returns **204 No Content** if the deletion is successful or the requested bucket policy does not exist.

Restrictions

- To delete the policy of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:DeleteBucketPolicy** in IAM or **DeleteBucketPolicy** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.deleteBucketPolicy(params)
```

Request Parameters

Table 5-65 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-66 Responses

Type	Description
Table 5-67 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-67 .

Table 5-67 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-68 .
InterfaceResult	Table 5-69	Explanation: Results outputted for a successful call. For details, see Table 5-69 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-68 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-69 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example deletes the policy of bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketPolicy() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Delete a bucket policy.
    const result = await obsClient.deleteBucketPolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s policy successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("Delete bucket(%s)'s policy fail!", params.Bucket);
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
deleteBucketPolicy();
```

Helpful Links

- [Deleting a Bucket Policy](#)
- [OBS Error Codes](#)
- [Access Control FAQ](#)

5.11 Obtaining the Region of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API returns the region of a bucket.

Restrictions

- To obtain the region of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketLocation** in IAM or **GetBucketLocation** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketLocation(params)
```

Request Parameters

Table 5-70 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-71 Responses

Type	Description
Table 5-72 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-72 .

Table 5-72 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-73 .
InterfaceResult	GetBucketMetadataOutput	Explanation: Results outputted for a successful call. For details, see Table 5-74 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-73 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-74 GetBucketLocationOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Location	string	Explanation: Region where a bucket is located. Restrictions: If the used endpoint is obs.myhuaweicloud.com , this parameter is not required. If any other endpoint is used, this parameter is required. Value range: To learn about valid regions and endpoints, see Regions and Endpoints . An endpoint is the request address for calling an API. Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator. Default value: If obs.myhuaweicloud.com is used as the endpoint and no region is specified, cn-north-1 (the CN North-Beijing1 region) is used by default.

Code Examples

You can call **ObsClient.getBucketLocation** to obtain the location of a bucket. This example uses bucket **examplebucket**.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.
```



```
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLocation() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Obtain the bucket region.
    const result = await obsClient.getBucketLocation(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s location successful!", params.Bucket);
      console.log("Location: %s", result.InterfaceResult.Location);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getBucketLocation();
```

NOTE

When creating a bucket, you can specify its location. For details, see [Creating a Bucket \(SDK for Node.js\)](#).

5.12 Obtaining Storage Information of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API returns the storage information on a bucket, including the number of objects and the space occupied by the objects in the bucket.

 NOTE

OBS measures bucket storage statistics in the background and does not update the storage information in real time. So, you are advised not to perform real-time verification on the storage information.

Restrictions

- To obtain the storage information of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketStorage** in IAM or **GetBucketStorage** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketStorageInfo(params)
```

Request Parameters

Table 5-75 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-76 Responses

Type	Description
Table 5-77 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-77 .

Table 5-77 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-78 .
InterfaceResult	Table 5-79	Explanation: Results outputted for a successful call. For details, see Table 5-79 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-78 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-79 GetBucketStorageInfoOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Size	number	Explanation: Size of the space occupied by objects in the bucket.
ObjectNumber	number	Explanation: Number of objects in the bucket.

Code Examples

A bucket's storage information comprises its used size and object count. You can call **ObsClient.getBucketStorageInfo** to obtain the bucket storage information. This example uses bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketStorageInfo() {
  try {
    const params = {
      // Specify the bucket name.
```

```
    Bucket: "examplebucket"
  };
  // Obtain the bucket storage information.
  const result = await obsClient.getBucketStorageInfo(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Get bucket(%s)'s storage-info successful!", params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log('Size: %s', result.InterfaceResult.Size);
    console.log('ObjectNumber: %s', result.InterfaceResult.ObjectNumber);
    return;
  }

  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketStorageInfo();
```

5.13 Configuring a Storage Quota for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

A quota limits the maximum capacity allowed in a bucket. By default, there is no limit on the storage capacity of the entire OBS system or a single bucket, and any number of objects can be stored. You can set a storage quota to control the total size of objects that can be uploaded to the bucket. If the storage quota is reached, object upload will fail.

A quota limit does not apply to the objects uploaded before the quota is configured. If the specified quota is already smaller than the total size of existing objects in the bucket, the existing objects in the bucket will not be deleted, but no more object can be uploaded to the bucket later. In this case, you can upload new objects only by deleting some existing objects until the used space is less than the quota.

Restrictions

- A bucket storage quota must be a non-negative integer expressed in bytes. The maximum value is $2^{63} - 1$.
- OBS does not provide an API for deleting bucket storage quotas. You can set the bucket storage quota to **0** to cancel the limit.

- To configure a storage quota for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketQuota** in IAM or **PutBucketQuota** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketQuota(params)
```

Request Parameters

Table 5-80 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
StorageQuota	number	Yes	Explanation: Bucket quota. Restrictions: None Value range: 0 to $(2^{63} - 1)$, in bytes Default value: 0, indicating that there is no limit on the bucket quota.

Responses

Table 5-81 Responses

Type	Description
Table 5-82 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-82 .

Table 5-82 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-83 .
InterfaceResult	Table 5-84	Explanation: Results outputted for a successful call. For details, see Table 5-84 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-83 ICommonMsg

Parameter	Type	Description

Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-84 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.setBucketQuota** to configure a storage quota for a bucket. This example sets the quota of bucket **examplebucket** to 1 GB.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
  //coding may result in leakage.  
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,
```

```
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketQuota() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify a quota of 1 GB (measured in bytes) for the bucket.
      StorageQuota: 1024 * 1024 * 1024
    };
    // Set the bucket quota.
    const result = await obsClient.setBucketQuota(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s quota successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setBucketQuota();
```

NOTE

- Use the **StorageQuota** parameter to specify the bucket quota.
- A bucket quota must be a non-negative integer expressed in bytes. The maximum value is $2^{63} - 1$.

5.14 Obtaining the Storage Quota of a Bucket (SDK for Node.js)

Function

This API returns the storage quota (upper limit of the storage capacity) of a bucket. If the quota is 0, there is no upper limit on the bucket capacity.

Restrictions

- A bucket storage quota must be a non-negative integer expressed in bytes. The maximum value is $2^{63} - 1$.
- A bucket owner with a frozen account in arrears is not allowed to query the bucket storage quota.
- To obtain the storage quota of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketQuota** in IAM or

GetBucketQuota in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketQuota(params)
```

Request Parameters

Table 5-85 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-86 Responses

Type	Description
Table 5-87 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-87 .

Table 5-87 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-88 .
InterfaceResult	Table 5-89	Explanation: Results outputted for a successful call. For details, see Table 5-89 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-88 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-89 GetBucketQuotaOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
StorageQuota	number	Explanation: Bucket quota. Default value: 0 , indicating that there is no limit on the bucket quota.

Code Examples

You can call **ObsClient.getBucketQuota** to obtain the storage quota of a bucket. This example uses bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketQuota() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    }
  }
}
```

```
};  
// Obtain the bucket quota.  
const result = await obsClient.getBucketQuota(params);  
if (result.CommonMsg.Status <= 300) {  
  console.log("Get bucket(%s)'s quota successful!", params.Bucket);  
  console.log("RequestId: %s", result.CommonMsg.RequestId);  
  console.log("StorageQuota: %s", result.InterfaceResult.StorageQuota);  
  return;  
}  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
}  
};  
};  
  
getBucketQuota();
```

5.15 Configuring a Storage Class for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API configures a storage class for a bucket. If you upload or copy objects to a bucket with a storage class configured or initiating a multipart upload for such a bucket without specifying storage classes for objects, those objects inherit the storage class of the bucket.

Restrictions

- To configure a storage class for a bucket, you must be the bucket owner or have the required permission (**obs:PutBucketStoragePolicy** in IAM or **PutBucketStoragePolicy** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketStoragePolicy(params)
```


Request Parameters

Table 5-90 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	Yes	Explanation: Storage class of the bucket. Restrictions: None Value range: See Table 5-91 . Default value: None

Table 5-91 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 5-92 Responses

Type	Description
Table 5-93 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-93 .

Table 5-93 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-94 .
InterfaceResult	Table 5-95	Explanation: Results outputted for a successful call. For details, see Table 5-95 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-94 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-95 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.setBucketStoragePolicy** to specify the storage class for a bucket. This example sets the storage class of bucket **examplebucket** to Infrequent Access.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketStoragePolicy() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify a storage class (obsClient.enums.StorageClassWarm in this example) for the bucket.
      StorageClass: obsClient.enums.StorageClassWarm
    };
    // Configure a storage class for the bucket.
    const result = await obsClient.setBucketStoragePolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s storage-class successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    }
  } catch (error) {
    console.error(error);
  }
}
```

```
        return;  
    };  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
    } catch (error) {  
        console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
        console.log(error);  
    };  
};  
};  
  
setBucketStoragePolicy();
```

5.16 Obtaining the Storage Class of a Bucket (SDK for Node.js)

Function

This API returns the storage class of a bucket.

Restrictions

- To obtain a bucket's storage class, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketStoragePolicy** in IAM or **GetBucketStoragePolicy** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketStoragePolicy(params)
```

Request Parameters

Table 5-96 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 5-97 Responses

Type	Description
Table 5-98 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 5-98 .

Table 5-98 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 5-99 .
InterfaceResult	GetBucketMetadataOutput	Explanation: Results outputted for a successful call. For details, see Table 5-100 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 5-99 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 5-100 GetBucketStorageClassOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
StorageClass	StorageClass Type	Explanation: Storage class of the bucket. Value range: See Table 5-101 .

Table 5-101 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

You can call **ObsClient.getBucketStoragePolicy** to obtain the storage class of a bucket. This example uses bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketStoragePolicy() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Obtain the storage class of the bucket.
    const result = await obsClient.getBucketStoragePolicy(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s storage-class successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass: %s", result.InterfaceResult.StorageClass);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getBucketStoragePolicy();
```

Helpful Links

- [Obtaining the Storage Class of a Bucket](#)
- [OBS Error Codes](#)
- [FAQ for Buckets and Objects](#)

6 Object Upload (SDK for Node.js)

6.1 Object Upload Overview (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

In OBS, objects are basic data units that users can perform operations on. OBS Node.js SDK provides abundant APIs for object upload in the following methods:

- [Uploading an Object - Text-Based \(SDK for Node.js\)](#)
- [Uploading an Object - Streaming \(SDK for Node.js\)](#)
- [Uploading an Object - File-Based \(SDK for Node.js\)](#)
- [Initiating a Multipart Upload \(SDK for Node.js\)](#)
- [Uploading an Object - Append \(SDK for Node.js\)](#)
- [Uploading an Object - Resumable \(SDK for Node.js\)](#)
- [Uploading an Object - Browser-Based \(SDK for Node.js\)](#)

The SDK supports the upload of objects whose size ranges from 0 KB to 5 GB. For streaming upload, appendable upload, and file-based upload, data to be uploaded at a time cannot be larger than 5 GB. If the file is larger than 5 GB, multipart upload (whose part size is smaller than 5 GB) is suitable. Browser-based upload allows files to be uploaded through a browser.

If you grant anonymous users the read permission for an object during the upload, anonymous users can access the object through a URL after the upload is complete. The object URL is in the format of **`https://bucket.name.domain.name/directory levels/object name`**. If the object resides in the root directory of the bucket, its URL does not contain directory levels.

6.2 Uploading an Object - Text-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads an object to a bucket.

Restrictions

- To upload an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, refer to [Multipart Upload APIs \(SDK for Node.js\)](#).

Method

```
ObsClient.putObject(params)
```

Request Parameters

Table 6-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs . Restrictions: None Value range: See Table 6-2 . Default value: None
StorageClass	StorageClassType	No	Explanation: When creating an object, you can use this header to specify the storage class for the object. Restrictions: None Value range: See Table 6-3 . Default value: If this parameter is not specified, the object inherits the storage class of the bucket.

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Content of the part to upload, which can be in string or stream.Readable form. Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). • Body and SourceFile cannot be used together. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: • The content size in a single upload ranges from 0 to 5 GB. • Body and SourceFile cannot be used together. • Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: None Default value: None
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress Restrictions: None Value range: None Default value: None NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds).
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58lG6hfM7vql4K0vnWpog== Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirect-Location	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify Content-Type when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to Content-Type.

Parameter	Type	Mandatory (Yes/No)	Description
ContentLength	int64	No	Explanation: Size of the object to be uploaded. Restrictions: - The object size in a single upload ranges from 0 to 5 GB. - To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Value range: 0 GB to 5 GB Default value: If this parameter is not specified, the SDK automatically calculates the size of the object.
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: - If this parameter is not specified, the default master key will be used. - If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: - The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. - This parameter can only be configured when uploading an object. Value range: 1 to ($2^{63} - 1$), in days Default value: None

Table 6-2 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 6-3 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-4 Responses

Type	Description
Table 6-5 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-5 .

Table 6-5 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-6 .

Parameter	Type	Description
InterfaceResult	Table 6-7	Explanation: Results outputted for a successful call. For details, see Table 6-7 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-6 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-7 PutObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Object storage class. Value range: - If the storage class of the object is Standard, this parameter is left blank. - See Table 6-8.
VersionId	string	Explanation: Object version. If versioning is enabled for the bucket, the object version ID will be returned.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Table 6-8 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

Text-based upload is used to upload character strings. You can call **ObsClient.putObject** to upload character strings to OBS. This example uploads a text object to bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
```

```
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify a text object.
      Body: 'Hello OBS'
    };
    // Upload the object.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass: %s, ETag: %s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

putObject();
```

6.3 Uploading an Object - Streaming (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads an object to a bucket.

Restrictions

- To upload an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, refer to [Multipart Upload APIs \(SDK for Node.js\)](#).

Method

```
ObsClient.putObject(params)
```

Request Parameters

Table 6-9 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.) – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs . Restrictions: None Value range: See Table 6-10 . Default value: None
StorageClass	StorageClassType	No	Explanation: When creating an object, you can use this header to specify the storage class for the object. Restrictions: None Value range: See Table 6-11 . Default value: If this parameter is not specified, the object inherits the storage class of the bucket.

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Content of the part to upload, which can be in string or stream.Readable form. Restrictions: - The object size in a single upload ranges from 0 to 5 GB. - To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Body and SourceFile cannot be used together. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: - The content size in a single upload ranges from 0 to 5 GB. Body and SourceFile cannot be used together. Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: None Default value: None
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress Restrictions: None Value range: None Default value: None NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds).
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58IG6hfM7vql4K0vnWpog== Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirect-Location	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify Content-Type when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to Content-Type.

Parameter	Type	Mandatory (Yes/No)	Description
ContentLength	int64	No	Explanation: Size of the object to be uploaded. Restrictions: - The object size in a single upload ranges from 0 to 5 GB. - To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Value range: 0 GB to 5 GB Default value: If this parameter is not specified, the SDK automatically calculates the size of the object.
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: • <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. • <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). • <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: • If this parameter is not specified, the default master key will be used. • If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: - The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. - This parameter can only be configured when uploading an object. Value range: 1 to ($2^{63} - 1$), in days Default value: None

Table 6-10 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 6-11 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-12 Responses

Type	Description
Table 6-13 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-13 .

Table 6-13 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-14 .

Parameter	Type	Description
InterfaceResult	Table 6-15	Explanation: Results outputted for a successful call. For details, see Table 6-15 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-14 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-15 PutObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Object storage class. Value range: - If the storage class of the object is Standard, this parameter is left blank. - See Table 6-16.
VersionId	string	Explanation: Object version. If versioning is enabled for the bucket, the object version ID will be returned.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Table 6-16 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples: Uploading a Network Stream

Streaming upload uses **stream.Readable** as the data source of objects. This example uploads an object to bucket **examplebucket** using a stream.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
```

```
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getNetWorkStream() {
  // Define the network stream URL.
  const url = 'https://www.example.com';
  // Import HTTP and HTTPS libraries.
  const http = require('http');
  const https = require('https');
  // Choose the corresponding library based on the URL.
  const request = url.startsWith('http') ? http : https;
  return new Promise((resolve, reject) => {
    request.get(url, res => {
      if (res.statusCode === 200) {
        resolve(res);
      } else {
        reject(res);
      }
    });
  });
}

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the network flow.
      Body: await getNetWorkStream()
    };
    // Upload the object using the network stream.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

putObject();
```

Code Examples: Uploading a File Stream

This example uploads an object to bucket **examplebucket** using a file stream.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
```

```
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getFileStream() {
  const fs = require('fs');
  // Enter the complete local path of the file to be uploaded.
  // If no local path is specified, the system uploads the file from the local path of the project that the
  // sample program belongs to by default.
  return fs.createReadStream('D:\\localpath\\examplefile.txt');
}

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the file stream.
      Body: await getFileStream()
    };
    // Upload the object using the file stream.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

putObject();
```

6.4 Uploading an Object - File-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads an object to a bucket.

Restrictions

- To upload an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, refer to [Multipart Upload APIs \(SDK for Node.js\)](#).

Method

```
ObsClient.putObject(params)
```

Request Parameters

Table 6-17 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs . Restrictions: None Value range: See Table 6-18 . Default value: None
StorageClass	StorageClassType	No	Explanation: When creating an object, you can use this header to specify the storage class for the object. Restrictions: None Value range: See Table 6-19 . Default value: If this parameter is not specified, the object inherits the storage class of the bucket.

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Content of the part to upload, which can be in string or stream.Readable form. Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). • Body and SourceFile cannot be used together. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: • The content size in a single upload ranges from 0 to 5 GB. • Body and SourceFile cannot be used together. • Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: None Default value: None
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress Restrictions: None Value range: None Default value: None NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds).
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58lG6hfM7vql4K0vnWpog== Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirect-Location	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify Content-Type when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to Content-Type.

Parameter	Type	Mandatory (Yes/No)	Description
ContentLength	int64	No	Explanation: Size of the object to be uploaded. Restrictions: - The object size in a single upload ranges from 0 to 5 GB. - To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Value range: 0 GB to 5 GB Default value: If this parameter is not specified, the SDK automatically calculates the size of the object.
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: • <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. • <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). • <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: • If this parameter is not specified, the default master key will be used. • If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: - The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. - This parameter can only be configured when uploading an object. Value range: 1 to ($2^{63} - 1$), in days Default value: None

Table 6-18 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 6-19 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-20 Responses

Type	Description
Table 6-21 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-21 .

Table 6-21 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-22 .

Parameter	Type	Description
InterfaceResult	Table 6-23	Explanation: Results outputted for a successful call. For details, see Table 6-23 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-22 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-23 PutObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Object storage class. Value range: - If the storage class of the object is Standard, this parameter is left blank. - See Table 6-24.
VersionId	string	Explanation: Object version. If versioning is enabled for the bucket, the object version ID will be returned.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Table 6-24 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

File-based upload uses local files as data sources. This example uploads a file to bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
```

```
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object. example/objectname is used in this example.
      Key: "example/objectname",
      // localfile indicates the path of the local file to be uploaded, which must include the file name.
      SourceFile: 'localfile'
    };
    // Upload the file.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

putObject();
```

6.5 Creating a Folder (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads a folder object to a bucket to make it easy to manage data objects.

Restrictions

- To upload an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- To create a folder in OBS is to create an object whose size is 0 and whose name ends with a slash (/).

Method

```
ObsClient.putObject(params)
```

Request Parameters

Table 6-25 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs . Restrictions: None Value range: See Table 6-26 . Default value: None
StorageClass	StorageClassType	No	Explanation: When creating an object, you can use this header to specify the storage class for the object. Restrictions: None Value range: See Table 6-27 . Default value: If this parameter is not specified, the object inherits the storage class of the bucket.

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Content of the part to upload, which can be in string or stream.Readable form. Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). • Body and SourceFile cannot be used together. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: • The content size in a single upload ranges from 0 to 5 GB. • Body and SourceFile cannot be used together. • Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: None Default value: None
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress Restrictions: None Value range: None Default value: None NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds).
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58lG6hfM7vql4K0vnWpog== Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify this parameter when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to this parameter.

Parameter	Type	Mandatory (Yes/No)	Description
ContentLength	int64	No	Explanation: Size of the object to be uploaded. Restrictions: - The object size in a single upload ranges from 0 to 5 GB. - To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Value range: 0 GB to 5 GB Default value: If this parameter is not specified, the SDK automatically calculates the size of the object.
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: • <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. • <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). • <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: • If this parameter is not specified, the default master key will be used. • If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: - The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. - This parameter can only be configured when uploading an object. Value range: 1 to ($2^{63} - 1$), in days Default value: None

Table 6-26 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 6-27 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-28 Responses

Type	Description
Table 6-29 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-29 .

Table 6-29 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-30 .

Parameter	Type	Description
InterfaceResult	Table 6-31	Explanation: Results outputted for a successful call. For details, see Table 6-31 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-30 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-31 PutObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Object storage class. Value range: - If the storage class of the object is Standard, this parameter is left blank. - See Table 6-32.
VersionId	string	Explanation: Object version. If versioning is enabled for the bucket, the object version ID will be returned.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Table 6-32 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

All elements stored in OBS buckets are called objects. A folder in OBS is essentially an object with a size of 0 and a name ending with a slash (/). You can perform download, delete, or other operations on folder objects just like on regular objects. This example creates a folder object named **parent_directory**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
});
```



```
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify a folder (parent_directory in this example).
      Key: "parent_directory/",
    };
    // Create a folder object.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

putObject();
```

NOTE

- To create a folder in OBS is to create an object whose size is 0 and whose name ends with a slash (/), in essential.

6.6 Configuring Object Metadata (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Object metadata contains a set of name-value pairs that are used for describing and managing objects.

Currently, only the system-defined metadata is supported. System-defined metadata consists of system-controlled metadata and user-controlled metadata.

This API sends a HEAD request for obtaining object metadata.

Restrictions

- To obtain object metadata, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Users with the READ permission for objects can send HEAD requests to retrieve object metadata, which is included in the response.

Method

```
ObsClient.setObjectMetadata(params)
```

Request Parameters

Table 6-33 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (.), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters.

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
MetadataDirective	MetadataDirectiveType	Yes	Explanation: Policy for copying the source object's attributes Restrictions: None Value range: See Table 6-39 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
VersionId	string	No	Explanation: Object version. Example: G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
CacheControl	string	No	Explanation: Cache-Control header in the response. It specifies the cache behavior of the web page when an object is downloaded. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ContentDisposition	string	No	Explanation: Content-Disposition header in the response. It specifies the name of an object when it is downloaded. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentEncoding	string	No	Explanation: Content-Encoding header in the response. It specifies the content encoding format when an object is downloaded. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ContentLanguage	string	No	Explanation: Content-Language header in the response. It specifies what language the object content is in when being downloaded. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ContentType	string	No	Explanation: Content-Type header in the response. It specifies the file type of an object when it is downloaded. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	string	No	Explanation: Expires header in the response. It specifies the cache expiration time of the web page when the object is downloaded. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation . Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	No	Explanation: Storage class of the object. Restrictions: None Value range: See Table 6-34 . Default value: None
Metadata	object	No	Explanation: Custom metadata of the object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Table 6-34 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-35 Responses

Type	Description
Table 6-36 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-36 .

Table 6-36 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-37 .

Parameter	Type	Description
InterfaceResult	Table 6-38	Explanation: Results outputted for a successful call. For details, see Table 6-38 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-37 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-38 SetObjectMetadataOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
MetadataDirective	Metadata Directive Type	Explanation: Policy for copying the source object's attributes Value range: See Table 6-39 .
CacheControl	string	Explanation: Cache-Control header in the response. It specifies the cache behavior of the web page when an object is downloaded.
ContentDisposition	string	Explanation: Content-Disposition header in the response. It specifies the name of an object when it is downloaded.
ContentEncoding	string	Explanation: Content-Encoding header in the response. It specifies how the object is encoded when being downloaded.
ContentLanguage	string	Explanation: Content-Language header in the response. It specifies what language the object content is in when being downloaded.
ContentType	string	Explanation: Content-Type header in the response. It specifies the file type of an object when it is downloaded.
Expires	string	Explanation: Expires header in the response. It specifies the cache expiration time of the web page when the object is downloaded.

Parameter	Type	Description
WebsiteRedirectLocation	string	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory.
StorageClass	StorageClassType	Explanation: Storage class configured when copying the object. If this parameter is not specified, the object inherits the storage class of the bucket. Value range: See Table 6-40.

Parameter	Type	Description
Metadata	object	Explanation: Custom metadata of the object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations.

Table 6-39 MetadataDirectiveType

Constant	Default Value	Description
ObsClient.enums.CopyMetadata	COPY	When copying an object, the object's attributes are also copied. NOTICE This value is used only in the API for Copying an Object (SDK for Node.js).
ObsClient.enums.ReplaceMetadata	REPLACE	REPLACE uses the complete header carried in the current request to replace the original one and deletes the metadata that is not specified.
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	REPLACE_NEW replaces the metadata that already has a value, assigns a value to the metadata that does not have a value, and retains the metadata that is not specified. NOTICE This value is used only in the API for Configuring Object Metadata (SDK for Node.js).

Table 6-40 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

This example configures metadata for object **example/objectname** in bucket **examplebucket**. The MIME type of the object is set to image/jpeg, the storage class is set to Archive, and custom metadata is also configured for the object.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectMetadata() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify a MIME type (image/jpeg in this example) for the object.
      ContentType: "image/jpeg",
    };
  }
}
```

```
// Specify a storage class (obsClient.enums.StorageClassCold in this example) for the object.  
StorageClass: obsClient.enums.StorageClassCold,  
// Specify custom metadata.  
Metadata: { "property1": "property-value1", "property2": "property-value2" }  
};  
// Configure the metadata for the object.  
const result = await obsClient.setObjectMetadata(params);  
if (result.CommonMsg.Status <= 300) {  
  console.log("Set Object(%)s's metadata successful with bucket(%)!", params.Key, params.Bucket);  
  console.log("RequestId: %s", result.CommonMsg.RequestId);  
  return;  
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
setObjectMetadata();
```

6.7 Configuring a Lifecycle Rule When Uploading an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

When uploading an object or initiating a multipart upload, you can set an expiration time for the object using **Expires**. This method only supports setting the object expiration time in days, and the expired objects will be automatically deleted by OBS, with a higher priority than bucket lifecycle rules.

Code Examples: Uploading an Object

When uploading an object, you can specify when it expires after being created.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
  coding may result in leakage.  
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
  access_key_id: process.env.ACCESS_KEY_ID,
```

```
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
as an example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object (example/objectname in this example).
      Key: "example/objectname",
      // localfile indicates the path of the local file to be uploaded, which must include the file name.
      SourceFile: "localfile",
      // Specify how many days can pass before the object expires (30 in this example).
      Expires: 30
    };
    // Upload a file.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

putObject()
```

Code Examples: Initiating a Multipart Upload

When initiating a multipart upload, you can specify when the object expires after it is created.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
as an example. Replace it with the one currently in use.
```



```
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function initiateMultipartUpload() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object (example/objectname in this example).
      Key: "example/objectname",
      // Specify how many days can pass before the object expires (30 in this example).
      Expires : 30
    };
    // Initiate the multipart upload.
    const result = await obsClient.initiateMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Initiate multipart upload successfully with bucket(%s) and object(%s)!", params.Bucket,
        params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("UploadId: %s", result.InterfaceResult.UploadId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
      response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
      attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

initiateMultipartUpload();
```

6.8 Uploading an Object - Append (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads a file or folder to an existing OBS bucket. You can upload text, pictures, videos, or any other types of files.

This API adds data to the end of a specified object. If there is no object with the same key found in the bucket, a new object is created.

The latest modification time of the object is updated each time an upload is appended.

Restrictions

- To upload an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy).

For details, see [Introduction to OBS Access Control, IAM Custom Policies, and Configuring an Object Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The name of each object in a bucket must be unique.
- If SSE-C is used for server-side encryption, you must carry request headers such as **x-obs-server-side-encryption** in each append upload.
- If SSE-KMS is used for server-side encryption, you only need to carry request headers such as **x-obs-server-side-encryption** when you first call this API to upload an object and when there is no existing object with the same name in the bucket.
- The size of each append upload cannot exceed 5 GB.
- The maximum number of append uploads for each appendable object is 10,000.
- If the storage class is COLD (Archive), this API cannot be called.
- If cross-region replication is configured for a bucket, this API cannot be called.
- Objects in parallel file systems are not appendable.
- Objects uploaded using **ObsClient.putObject**, referred to as normal objects, can overwrite objects uploaded using **ObsClient.appendObject**, referred to as appendable objects. Data cannot be appended to an appendable object anymore once the object has been overwritten by a normal object.
- When you upload an object for the first time in appendable mode, an exception will be reported (HTTP status code **409**) if a common object with the same name exists.
- The ETag returned for an appendable upload is the ETag for the uploaded content, rather than that of the whole object.

Method

```
ObsClient.appendObject(params)
```

Request Parameters

Table 6-41 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
Position	number	Yes	Explanation: Position where the object data is appended Restrictions: For an object to be appended, the value of position must be set to 0 when the object is uploaded for the first time. For the second append upload, the value of position should be set to the value of NextAppendPosition returned in the response when the first upload is successful. Value range: 0 to $(2^{63} - 1)$, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Content of the part to upload, which can be in string or stream.Readable form. Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). • Body and SourceFile cannot be used together. • If both Body and SourceFile are left blank, the size of the uploaded object is 0 bytes. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: • The content size in a single upload ranges from 0 to 5 GB. • Body and SourceFile cannot be used together. • If both Body and SourceFile are left blank, the size of the uploaded object is 0 bytes. • Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset and SourceFile are used together to specify what data is uploaded from the source file. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds). Restrictions: None Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58IG6hfM7vql4K0vnWpog== Default value: None
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs. Restrictions: This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: See Table 6-42. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify this parameter when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to this parameter.
ContentLength	int64	No	Explanation: Size of the object to be uploaded. Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, refer to Multipart Upload APIs (SDK for Node.js). Value range: 0 GB to 5 GB Default value: If this parameter is not specified, the SDK automatically calculates the size of the object.

Parameter	Type	Mandatory (Yes/No)	Description
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: • Currently, only KMS is supported. • This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Default value: - If this parameter is not specified, the default master key will be used. - If there is not a default master key, OBS will create one and use it.

Parameter	Type	Mandatory (Yes/No)	Description
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: - Only AES256 is supported. - This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: AES256 Default value: None
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: - A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= - This parameter can only be configured in the first request for append upload. The configurations specified in the first request will be used in subsequent requests by default. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	No	Explanation: When creating an object, you can use this header to specify the storage class for the object. Restrictions: None Value range: See Table 6-43 . Default value: If this parameter is not specified, the object inherits the storage class of the bucket.

Parameter	Type	Mandatory (Yes/No)	Description
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • This parameter is valid only for the first append upload. • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: • The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. • This parameter can only be configured when uploading an object. Value range: 1 to $(2^{63} - 1)$, in days Default value: None

Table 6-42 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 6-43 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 6-44 Responses

Type	Description
Table 6-45 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-45 .

Table 6-45 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-46 .

Parameter	Type	Description
InterfaceResult	Table 6-47	Explanation: Results outputted for a successful call. For details, see Table 6-47 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-46 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-47 AppendObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
VersionId	string	Explanation: Object version. If versioning is enabled for the bucket, the object version ID will be returned.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
NextAppendPosition	int64	Explanation: Position from which the next append upload starts Restrictions: For an object to be appended, the value of position must be set to 0 when the object is uploaded for the first time. For the second append upload, the value of position should be set to the value of NextAppendPosition returned in the response when the first upload is successful. You can also call ObsClient.getObjectMetadata to obtain the value of NextAppendPosition .
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Code Examples

You can call **ObsClient.appendObject** to upload an object and append content to it. This example uses object **example/objectname** and bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function appendObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: "example/objectname",
      // Specify the position where content is appended. For the first upload, set Position to 0. For the
      // subsequent appends, specify the actual position.
      Position: 0,
      // Specify the data stream of the object to upload.
      Body: strings.NewReader("Hello OBS"),
    };
    // Upload the object using the append method.
    const result = await obsClient.appendObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Append object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s, NextAppendPosition:%d", result.InterfaceResult.ETag,
        result.InterfaceResult.NextAppendPosition);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
appendObject();
```

6.9 Uploading an Object - Resumable (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

The resumable upload is an encapsulated and enhanced version of [the multipart upload](#) used for dealing with possible upload failures of large files when the network connection is unstable or a program crashes.

Uploading large files often fails due to poor network conditions or program breakdowns. It is a waste of resources to restart the upload process upon an upload failure, and the restarted upload process may still suffer from the unstable network. To resolve such issues, you can use the API for resumable upload, whose working principle is to divide the to-be-uploaded file into multiple parts and upload them separately. The upload result of each part is recorded in a checkpoint file in real time. Only when all parts are successfully uploaded, the result indicating a successful upload will be returned. Otherwise, an error is returned in callback function to remind you of calling the API again for re-uploading. Based on the upload status of each part recorded in the checkpoint file, the re-uploading will upload the parts failed to be uploaded previously, instead of uploading all parts. By virtue of this, resources are saved and efficiency is improved.

The resumable upload interface helps save resources and improve efficiency by restarting an upload from the point of failure and concurrently uploading parts. You do not need to worry about internal service details, such as the creation and deletion of checkpoint files, division of objects, or concurrent uploads of parts.

Restrictions

The total size of files uploaded by the resumable upload API must be larger than 100 KB.

Method

```
ObsClient.uploadFile(params)
```


Request Parameters

Table 6-48 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
UploadFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: None Value range: None Default value: None
PartSize	number	No	Explanation: Size of the current part. Restrictions: None Value range: The value ranges from 100 KB to 5 GB, in bytes. Default value: 102400

Parameter	Type	Mandatory (Yes/No)	Description
TaskNum	number	No	Explanation: Maximum number of parts that can be uploaded concurrently Restrictions: None Value range: 1~10000 Default value: 1, indicating concurrent uploads are not used
ProgressCallback	function	No	Explanation: Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds). Restrictions: None Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResumeCallback	function	No	Explanation: Callback function used to obtain the control parameter for canceling a resumable download Restrictions: None Value range: None Default value: None NOTE • This callback function contains a control parameter used for pausing or aborting a resumable upload. • By calling the cancel method of this control parameter, you can pause a resumable upload. • By calling the abort method of this control parameter, you can abort a resumable upload.
EnableCheckpoint	boolean	No	Explanation: Whether to enable the resumable mode. Restrictions: None Value range: • true: The resumable mode is enabled. • false: The resumable mode is disabled. In this case, this API works as a multipart upload API, and no checkpoint files are generated. Default value: false

Parameter	Type	Mandatory (Yes/No)	Description
CheckpointFile	string	No	Explanation: Path of a file generated for recording the progress of a resumable upload. The file contains the information about parts and progress. Restrictions: This parameter is valid only in the resumable mode. Value range: None Default value: If this parameter is left blank, the checkpoint file will be in the same directory as the local file to be uploaded.
EnableCheckSum	boolean	No	Explanation: Whether to verify the file to upload. If this function is enabled, before each task restarts, the system verifies whether the file to upload is the one used during task initialization. Restrictions: This parameter is valid only in the resumable mode. Value range: • true: The file is verified. • false: The file is not verified. Default value: false

Parameter	Type	Mandatory (Yes/No)	Description
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify this parameter when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to this parameter.
ACL	AclType	No	Explanation: Access control list (ACL) that can be pre-defined when an object is created. For details about ACLs, see ACLs. Restrictions: None Value range: See Table 6-49. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirect-Location	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: - The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. - OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: - If this parameter is not specified, the default master key will be used. - If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes if used as a request parameter	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVa obncnLht/rCB2o/9Cw= Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Table 6-49 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Responses

Table 6-50 Responses

Type	Description
Table 6-51 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 6-51 .

Table 6-51 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 6-52 .
InterfaceResult	Table 6-53	Explanation: Results outputted for a successful call. For details, see Table 6-53 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 6-52 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 6-53 UploadFileOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an assembled object calculated based on the ETag of each part. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and B when the object is downloaded, this indicates the contents of the object have been changed. The ETag reflects changes to an object's contents, not its metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
Bucket	string	Explanation: Name of the bucket in which parts are assembled

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
Location	string	Explanation: URL of the object obtained from assembling the parts. Example: https://example-Bucket.obs.regions.myhuaweicloud.com/example-Object
VersionId	string	Explanation: Version ID of the object obtained from assembling the parts. If versioning is enabled for the bucket, the object version ID will be returned.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Code Examples

This example uploads **example/objectname** to **examplebucket** using resumable upload.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function uploadFile() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: 'examplebucket',
      // Specify the object to be created (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the local file to upload (/tmp/objectname in this example).
      UploadFile: 'localfile',
      // Specify whether to enable resumable transmission. Value true is used in this example. The default
      // value is false.
      EnableCheckpoint: true,
      // Specify a part size, in bytes. This example sets each part to 9 MB.
      PartSize: 9 * 1024 * 1024,
      // Specify the maximum number of parts that can be concurrently transmitted. 5 is used in this example.
      TaskNum: 5
    };
    // Upload the object using resumable upload.
    const result = await obsClient.uploadFile(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Upload file(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s", result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

uploadFile();
```


6.10 Uploading an Object - Browser-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Performing a browser-based upload is to upload objects to a specified bucket in HTML form. The maximum size of an object is 5 GB.

You can call **ObsClient.createPostSignatureSync** to generate request parameters for a browser-based upload. You can use Node.js code to simulate a browser-based upload. For details, see [post-object-sample](#). You can also perform a browser-based upload with the following steps:

- Step 1** Call **ObsClient.createPostSignatureSync** to generate request parameters for authentication.
- Step 2** Prepare an HTML form page.
- Step 3** Enter the request parameters in the HTML page.
- Step 4** Select a local file and upload it in browser-based mode.

----End

Method

```
ObsClient.createPostSignatureSync(params)
```

Request Parameters

Table 6-54 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
Expires	Number	No	Explanation: Validity period of authentication for a browser-based upload Restrictions: None Value range: A positive integer, in seconds Default value: 300

Parameter	Type	Mandatory (Yes/No)	Description
FormParams	Object	No	Explanation: Parameters used for browser-based uploads, excluding key , policy , and signature . Restrictions: None Value range: • acl • cache-control • content-type • content-disposition • content-encoding • expires Default value: None

Responses

Table 6-55 Responses

Parameter	Type	Description
OriginPolicy	String	Explanation: policy not encoded using Base64. This parameter can only be used for verification.
Policy	String	Explanation: policy in the form.
Signature	String	Explanation: signature in the form.

Code Examples

The following sample code shows how to generate the parameters in a browser-based upload request.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');
```

```
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

function createSignedUrlSync() {
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket",
    // Specify an object (example/objectname in this example).
    Key: "example/objectname",
    // Specify the validity period of the signed URL, in seconds (3600 in this example).
    Expires: 3600,
    // Specify the headers that must be carried in the request.
    FormParams: {
      'x-obs-acl': 'public-read',
      'content-type': 'text/plain'
    }
  };
  // Create a signed URL for uploading an object.
  const res = obsClient.createPostSignatureSync(params);
  console.log("OriginPolicy : %s", res.OriginPolicy);
  console.log("Policy: %s", res.Policy);
  console.log("Signature: %v", res.Signature);
};

createSignedUrlSync()
```

Code of an HTML form example is as follows:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>

<form action="http://bucketname.your-endpoint/" method="post" enctype="multipart/form-data">
Object key
<!-- Object name -->
<input type="text" name="key" value="objectname" />
<p>
ACL
<!-- Object ACL -->
<input type="text" name="x-obs-acl" value="public-read" />
<p>
Content-Type
<!-- Object MIME type -->
<input type="text" name="content-type" value="text/plain" />
<p>
<!-- Base64 code of the policy -->
<input type="hidden" name="policy" value="*** Provide your policy ***" />
<!-- AK -->
<input type="hidden" name="AccessKeyId" value="*** Provide your access key ***"/>
<!-- Signature information -->
<input type="hidden" name="signature" value="*** Provide your signature ***"/>
```

```
<input name="file" type="file" />
<input name="submit" value="Upload" type="submit" />
</form>
</body>
</html>
```

 NOTE

You can directly download the HTML form example [PostDemo](#).

7 Object Download (SDK for Node.js)

7.1 Overview (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

OBS Node.js SDK provides abundant APIs for downloading objects in the following modes:

- [Downloading an Object - Text-Based \(SDK for Node.js\)](#)
- [Downloading an Object - Streaming \(SDK for Node.js\)](#)
- [Downloading an Object - File-Based \(SDK for Node.js\)](#)
- [Downloading an Object - Range-Based \(SDK for Node.js\)](#)
- [Downloading an Object - Resumable \(SDK for Node.js\)](#)

You can call `ObsClient.getObject` to download an object.

7.2 Downloading an Object - Text-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads an object as text from OBS to your local computer.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```


Request Parameters

Table 7-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID. Example: G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: • true: The object is returned as a readable stream. • false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-2 Responses

Type	Description
Table 7-3 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-3 .

Table 7-3 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-4 .

Parameter	Type	Description
InterfaceResult	Table 7-5	Explanation: Results outputted for a successful call. For details, see Table 7-5 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-4 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-5 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class

Parameter	Type	Description
Expiration	string	Explanation: Expiration details
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a String object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example downloads object **examplebucket/objectname** as text.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
    };
    // Download the object as text.
```

```
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("Object Content: %s", result.InterfaceResult.Content);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getObject();
```

7.3 Downloading an Object - Streaming (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads an object as a stream from OBS to your local computer.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 7-6 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: true: The object is returned as a readable stream. false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-7 Responses

Type	Description
Table 7-8 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-8 .

Table 7-8 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-9 .

Parameter	Type	Description
InterfaceResult	Table 7-10	Explanation: Results outputted for a successful call. For details, see Table 7-10 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-9 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-10 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object.
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class

Parameter	Type	Description
Expiration	string	Explanation: Expiration details
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example downloads object **examplebucket/objectname** using the stream.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
      // Specify that the object is returned as a stream (stream.Readable).
      SaveAsStream: true
    };
  }
}
```

```
};

// Download the object using the stream.
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log('Object Content:');
  result.InterfaceResult.Content.on('data', (data) => {
    console.log(data.toString());
  });
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

getObject();
```

7.4 Downloading an Object - File-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads an object as a file from OBS to your local computer.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 7-11 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: • true: The object is returned as a readable stream. • false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-12 Responses

Type	Description
Table 7-13 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-13 .

Table 7-13 Response

Parameter	Type	Description
CommonMsg	ICCommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-14 .

Parameter	Type	Description
InterfaceResult	Table 7-15	Explanation: Results outputted for a successful call. For details, see Table 7-15 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-14 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-15 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class

Parameter	Type	Description
Expiration	string	Explanation: Expiration details
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example downloads object **examplebucket/objectname** as a file.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the destination path for the downloaded object.
      SaveAsFile: 'localfile'
    };
  }
}
```

```
};  
  
// Download an object as a file.  
const result = await obsClient.getObject(params);  
if (result.CommonMsg.Status <= 300) {  
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);  
  console.log("RequestId: %s", result.CommonMsg.RequestId);  
  return;  
}  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
}  
};  
};  
  
getObject();
```

7.5 Downloading an Object - Range-Based (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads part of an object from OBS to your local computer. If the specified range is from 0 to 1,000, data from byte 0 to byte 1,000, 1,001 bytes in total, are returned. If the specified range is invalid, the whole object will be returned. This download method can also be used to concurrently download parts of a large object. For details about the sample code, see [concurrent-download-object-sample](#).

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.
- If the specified range is invalid (because the start or end position is set to a negative integer or the range is larger than the object length), data of the whole object will be returned.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 7-16 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (.), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: true: The object is returned as a readable stream. false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y (bytes=1-1000 as an example) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images. Default value: If no commands are entered, the original image is returned.

Parameter	Type	Mandatory (Yes/No)	Description
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-17 Responses

Type	Description
Table 7-18 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-18 .

Table 7-18 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-19 .
InterfaceResult	Table 7-20	Explanation: Results outputted for a successful call. For details, see Table 7-20 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-19 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-20 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object.
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.

Parameter	Type	Description
Restore	string	Explanation: Restoration status of the object in the Archive storage class
Expiration	string	Explanation: Expiration details
Content	string stream.Readable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example downloads object **examplebucket/objectname** from 0 to 1,000 bytes.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
```

```
const params = {
  // Specify the bucket name.
  Bucket: "examplebucket",
  // Specify the object to download (example/objectname in this example).
  Key: 'example/objectname',
  // Specify the download range.
  Range: 'bytes=0-1000'
};
// Download the object based on the range.
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log('Object Content: %s', result.InterfaceResult.Content);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getObject();
```

7.6 Downloading an Object - Conditional (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads objects that meet specified conditions from OBS to a local computer.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 7-21 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: • true: The object is returned as a readable stream. • false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress Restrictions: None Value range: None Default value: None NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second).
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-22 Responses

Type	Description
Table 7-23 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-23 .

Table 7-23 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-24 .

Parameter	Type	Description
InterfaceResult	Table 7-25	Explanation: Results outputted for a successful call. For details, see Table 7-25 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-24 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-25 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class

Parameter	Type	Description
Expiration	string	Explanation: Expiration details
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example downloads object **examplebucket/objectname** and specifies that the object can be downloaded only when it has been modified since **Wed, 04 Jul 2018 08:54:53 GMT**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
```



```
// If modified since the specified time, the object is downloaded. Otherwise, a 304 code without a
message body is returned.
IfModifiedSince: 'Wed, 04 Jul 2018 08:54:53 GMT'
};
// Download the object based on the specified conditions.
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("Object Content: %s", result.InterfaceResult.Content);
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObject();
```

7.7 Rewriting Response Headers (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

When downloading an object, you can rewrite some HTTP/HTTPS response headers. The following table lists rewritable response headers.

Parameter	Description
ResponseContentType	Rewrites Content-Type in HTTP/HTTPS responses.
ResponseContentLanguage	Rewrites Content-Language in HTTP/HTTPS responses.
ResponseExpires	Rewrites Expires in HTTP/HTTPS responses.
ResponseCacheControl	Rewrites Cache-Control in HTTP/HTTPS responses.
ResponseContentDisposition	Rewrites Content-Disposition in HTTP/HTTPS responses.
ResponseContentEncoding	Rewrites Content-Encoding in HTTP/HTTPS responses.

Code Examples

This code example rewrites response headers.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
      // Rewrite a response header (Content-Type in this example).
      ResponseContentType: 'image/jpeg'
    };

    // Rewrite the response header.
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // Obtain the response header that was rewritten.
      console.log("ContentType:%s", result.InterfaceResult.ContentType);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getObject();
```

7.8 Downloading an Archive Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

CAUTION

To prolong the validity period of the Archive data restored, you can repeatedly restore the data, but you will be billed for each restoration. After a second restore, the validity period of Standard object copies will be prolonged, and you need to pay for storing these copies during the prolonged period.

Function

To download an object in the Archive storage class, you need to restore it first. After an object is restored, a copy of the object is saved in the Standard storage class. By doing so, the object in the Archive storage class and its copy in the Standard storage class co-exist in the bucket. The copy will be automatically deleted once its retention period ends.

This API restores an Archive object in a specified bucket.

Restrictions

- To restore an Archive object, you must be the bucket owner or have the required permission (**obs:object:RestoreObject** in IAM or **RestoreObject** in a bucket policy.) For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- To extend the validity period of the Archive data restored, you can repeatedly restore the data, but you will be billed for each restoration. After a restoration, the validity period of Standard object copies will be extended, and you need to pay for storing these copies during the extended period.

Method

```
ObsClient.restoreObject(params)
```

Request Parameters

Table 7-26 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: The object specified in ObsClient.restoreObject must be in the Archive storage class. Otherwise, an error is reported. Value range: The value can contain 1 to 1,024 characters. Default value: None
Version Id	string	No	Explanation: Version ID of the Archive object to restore. Restrictions: None Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is specified.

Parameter	Type	Mandatory (Yes/No)	Description
Days	number	Yes	Explanation: After an object is restored, a Standard copy is generated for the object. This parameter specifies how long the Standard copy can be retained, that is, the validity period of the restored object. Restrictions: None Value range: The value ranges from 1 to 30, in days. Default value: None
Tier	RestoreTierType	No	Explanation: Tier of the restoration speed. You can select a suitable tier based on your needs. Restrictions: None Value range: See Table 7-27 . Default value: Standard

Table 7-27 RestoreTierType

Constant	Default Value	Description
ObsClient.enums.RestoreTierExpedited	Expedited	Objects can be quickly restored from Archive storage within 1 to 5 minutes.
ObsClient.enums.RestoreTierStandard	Standard	Objects can be restored from Archive storage within 3 to 5 hours.

Responses

Table 7-28 Responses

Type	Description
Table 7-29 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-29 .

Table 7-29 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-30 .
InterfaceResult	Table 7-31	Explanation: Results outputted for a successful call. For details, see Table 7-31 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-30 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-31 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.restoreObject** to restore an Archive object. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function downloadColdObject() {
  // Specify the bucket name.
  const bucketName = "examplebucket";
  // Specify the object (example/objectname in this example).
  const objectKey = "example/objectname";
  try {
    // Restore an Archive object.
    const restoreObjectOutput = await obsClient.restoreObject({
      Bucket: bucketName,
      Key: objectKey,
      // Specify how long the restored object will be retained, in days. The value ranges from 1 to 30 (1 this
      // example).
      Days: 1,
      // Specify the restoration speed (obs.RestoreTierExpedited in this example). By default, the object is
      // restored at an expedited speed.
    });
  } catch (err) {
    console.log(err);
  }
}
```



```
Tier: obs.enums.RestoreTierExpedited
});
if (restoreObjectOutput.CommonMsg.Status > 300) {
  handleMessage(restoreObjectOutput.CommonMsg);
  return;
};
// Wait for the object to be restored.
await sleep(5 * 60);
// Download the object.
const getObjectOutput = await obsClient.getObject({
  Bucket: bucketName,
  Key: objectKey,
});
if (getObjectOutput.CommonMsg.Status > 300) {
  handleMessage(getObjectOutput.CommonMsg);
  return;
};
console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
console.log("RequestId: %s", result.CommonMsg.RequestId);
console.log('Object Content: %s', result.InterfaceResult.Content);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

function handleMessage(commonMsg) {
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", commonMsg.Status);
  console.log("Code: %s", commonMsg.Code);
  console.log("Message: %s", commonMsg.Message);
  console.log("RequestId: %s", commonMsg.RequestId);
};

function sleep() {
  return new Promise(resolve => {
    setTimeout(resolve, time * 1000);
  });
};

downloadColdObject();
```

7.9 Downloading an Object - Resumable (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

The API for resumable download is an encapsulated and enhanced version of range-based download. Downloading large files often fails due to an unstable network or program breakdown. It is a waste of resources to restart the download process upon a download failure, and the restarted download process may still suffer from the unstable network. To resolve such issues, you can use the API for resumable download, whose working principle is to divide the to-be-downloaded

file into multiple parts and download them separately. The download result of each part is recorded in a checkpoint file in real time. Only when all parts are successfully downloaded, the result indicating a successful download will be returned. Otherwise, an error is returned in callback function to remind you of calling the API again for re-downloading. Based on the download status of each part recorded in the checkpoint file, the re-downloading will download the parts failed to be downloaded previously, instead of downloading all parts. By virtue of this, resources are saved and efficiency is improved.

By resuming a failed download from where it failed, this API helps save resources. In addition, parts can be downloaded concurrently, which helps speed up the download. During the download process, you do not need to take care of internal service details, such as the creation and deletion of checkpoint files, division of objects, or concurrent downloads of parts.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.downloadFile(params)
```

Request Parameters

Table 7-32 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
DownloadFile	string	No	Explanation: The full local path that the object is downloaded to. If this parameter is not specified, the working directory of the current program is used. Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None
PartSize	number	No	Explanation: Size of the current part. Restrictions: None Value range: The value ranges from 100 KB to 5 GB, in bytes. Default value: 102400
TaskNum	number	No	Explanation: Maximum number of parts that can be downloaded concurrently Restrictions: None Value range: 1~10000 Default value: 1, indicating concurrent downloads are not used.

Parameter	Type	Mandatory (Yes/No)	Description
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
ResumeCallback	function	No	Explanation: Callback function used to obtain the control parameter for canceling a resumable download Restrictions: None Value range: None Default value: None NOTE - This callback function contains a control parameter used for canceling resumable downloads. - By calling the cancel method of this control parameter, you can pause a resumable download.

Parameter	Type	Mandatory (Yes/No)	Description
EnableCheckpoint	boolean	No	Explanation: Whether to enable the resumable mode. Restrictions: None Value range: • true: The resumable mode is enabled. • false: The resumable mode is disabled. In this case, this API works as a multipart download API, and no checkpoint files are generated. Default value: false
CheckpointFile	string	No	Explanation: Path of a file generated for recording the progress of a resumable download. The file contains the information about parts and progress. Restrictions: This parameter is valid only in the resumable mode. Value range: None Default value: If this parameter is left blank, the checkpoint file will be in the same directory as the local path that the file is downloaded to.

Parameter	Type	Mandatory (Yes/No)	Description
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-33 Responses

Type	Description
Table 7-34 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-34 .

Table 7-34 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-35 .
InterfaceResult	Table 7-36	Explanation: Results outputted for a successful call. For details, see Table 7-36 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-35 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .

Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-36 DownloadFileOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
CacheControl	string	Explanation: Cache-Control header in the response
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object

Parameter	Type	Description
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class
Expiration	string	Explanation: Expiration details
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeYMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Code Examples

This example downloads **example/objectname** from **examplebucket** using resumable download.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function downloadFile() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: 'examplebucket',
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
      // Specify a local absolute path (/tmp/objectname in this example) for download. If the path is left
      // blank, the current working directory is used by default.
      DownloadFile: 'localfile',
      // Specify whether to enable resumable transmission. Value true is used in this example. The default
      // value is false.
      EnableCheckpoint: true,
      // Specify a part size, in bytes. This example sets each part to 9 MB.
      PartSize: 9 * 1024 * 1024,
      // Specify the maximum number of parts that can be concurrently transmitted. 5 is used in this example.
      TaskNum: 5
    };
    // Download the object using resumable download.
    const result = await obsClient.downloadFile(params);
```

```
if (result.CommonMsg.Status <= 300) {
  console.log("Download file(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("StorageClass:%s, ETag:%s, ContentType:%s, ContentLength:%d, LastModified:%s",
    result.InterfaceResult.StorageClass, result.InterfaceResult.ETag, result.InterfaceResult.ContentType,
    result.InterfaceResult.ContentLength, result.InterfaceResult.LastModified,
  );
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
downloadFile();
```

7.10 Processing an Image (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

OBS can be used to process images in a stable, secure, efficient, easy to use, and cost-efficient manner. If you are about to download images, you can specify **ImageProcess** to process images, such as cropping, resizing, or watermarking images; or converting the formats of images.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Image objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 7-37 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: • true: The object is returned as a readable stream. • false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Multiple image processing commands can be executed in cascading mode, that is, in sequence. Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.

Parameter	Type	Mandatory (Yes/No)	Description
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 7-38 Responses

Type	Description
Table 7-39 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 7-39 .

Table 7-39 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 7-40 .
InterfaceResult	Table 7-41	Explanation: Results outputted for a successful call. For details, see Table 7-41 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 7-40 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 7-41 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size
CacheControl	string	Explanation: Cache-Control header in the response
ContentDisposition	string	Explanation: Content-Disposition header in the response
ContentEncoding	string	Explanation: Content-Encoding header in the response
ContentLanguage	string	Explanation: Content-Language header in the response
ContentType	string	Explanation: MIME type of the object
Expires	string	Explanation: Expires header in the response
ETag	string	Explanation: ETag of the object
VersionId	string	Explanation: Object version ID
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.

Parameter	Type	Description
Restore	string	Explanation: Restoration status of the object in the Archive storage class
Expiration	string	Explanation: Expiration details
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

This example processes and downloads image **examplebucket/objectname**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
```

```
    Bucket: "examplebucket",
    // Specify the object to download (example/objectname in this example).
    Key: 'example/objectname',
    // Resize and then rotate the image.
    ImageProcess: 'image/resize,m_fixed,w_100,h_100/rotate,90'
  };

  // Download the processed image.
  const result = await obsClient.getObject(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    return;
  }
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

getObject();
```

8 Object Management (SDK for Node.js)

8.1 Configuring Object Metadata (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Object metadata contains a set of name-value pairs that are used for describing and managing objects.

Currently, system-defined and custom metadata are supported.

System-defined metadata consists of system-controlled metadata and user-controlled metadata. The kind of metadata like **Last-Modified** is controlled by the system and cannot be modified. However, the kind of metadata configured for objects such as **ContentLanguage** can be modified by calling APIs.

This API adds, modifies, or deletes metadata of objects in a bucket.

Restrictions

- To configure object metadata, you must be the bucket owner or have the required permission (**obs:object:ModifyObjectMetaData** in IAM or **ModifyObjectMetaData** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- If versioning is enabled for a bucket, you can set metadata for objects of the latest version but cannot set metadata for historical objects.
- You cannot set metadata for Archive objects.

Method

```
ObsClient.setObjectMetadata(params)
```

Request Parameters

Table 8-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters.

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
MetadataDirective	MetadataDirectiveType	Yes	Explanation: Policy for copying the source object's attributes Restrictions: None Value range: See Table 8-7 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
CacheControl	string	No	Explanation: Cache-Control header in the response. It specifies the cache behavior of the web page when an object is downloaded. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ContentDisposition	string	No	Explanation: Content-Disposition header in the response. It specifies the name of an object when it is downloaded. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentEncoding	string	No	Explanation: Content-Encoding header in the response. It specifies the content encoding format when an object is downloaded. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ContentLanguage	string	No	Explanation: Content-Language header in the response. It specifies what language the object content is in when being downloaded. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ContentType	string	No	Explanation: Content-Type header in the response. It specifies the file type of an object when it is downloaded. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	string	No	Explanation: Expires header in the response. It specifies the cache expiration time of the web page when the object is downloaded. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation . Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	No	Explanation: Storage class of the object. Restrictions: None Value range: See Table 8-2 . Default value: None
Metadata	object	No	Explanation: Custom metadata of the object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Table 8-2 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 8-3 Responses

Type	Description
Table 8-4 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-4 .

Table 8-4 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-5 .

Parameter	Type	Description
InterfaceResult	Table 8-6	Explanation: Results outputted for a successful call. For details, see Table 8-6 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-5 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-6 SetObjectMetadataOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
MetadataDirective	Metadata Directive Type	Explanation: Policy for copying the source object's attributes Value range: See Table 8-7 .
CacheControl	string	Explanation: Cache-Control header in the response. It specifies the cache behavior of the web page when an object is downloaded.
ContentDisposition	string	Explanation: Content-Disposition header in the response. It specifies the name of an object when it is downloaded.
ContentEncoding	string	Explanation: Content-Encoding header in the response. It specifies the content encoding format when an object is downloaded.
ContentLanguage	string	Explanation: Content-Language header in the response. It specifies what language the object content is in when being downloaded.
ContentType	string	Explanation: Content-Type header in the response. It specifies the file type of an object when it is downloaded.
Expires	string	Explanation: Expires header in the response. It specifies the cache expiration time of the web page when the object is downloaded.

Parameter	Type	Description
WebsiteRedirectLocation	string	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory.
StorageClass	StorageClassType	Explanation: Storage class configured when copying the object. If this parameter is not specified, the object inherits the storage class of the bucket. Value range: See Table 8-8.

Parameter	Type	Description
Metadata	object	Explanation: Custom metadata of the object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations.

Table 8-7 MetadataDirectiveType

Constant	Default Value	Description
ObsClient.enums.CopyMetadata	COPY	When copying an object, the object's attributes are also copied. NOTICE This value is used only in the API for Copying an Object (SDK for Node.js).
ObsClient.enums.ReplaceMetadata	REPLACE	REPLACE uses the complete header carried in the current request to replace the original one and deletes the metadata that is not specified.
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	REPLACE_NEW replaces the metadata that already has a value, assigns a value to the metadata that does not have a value, and retains the metadata that is not specified. NOTICE This value is used only in the API for Configuring Object Metadata (SDK for Node.js).

Table 8-8 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

This example configures metadata for object **example/objectname** in bucket **examplebucket**. The MIME type of the object is set to image/jpeg, the storage class is set to Archive, and custom metadata is also configured for the object.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectMetadata() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify a MIME type (image/jpeg in this example) for the object.
      ContentType: "image/jpeg",
    };
  }
}
```



```
// Specify a storage class (obsClient.enums.StorageClassCold in this example) for the object.  
StorageClass: obsClient.enums.StorageClassCold,  
// Specify custom metadata.  
Metadata: { "property1": "property-value1", "property2": "property-value2" }  
};  
// Configure the metadata for the object.  
const result = await obsClient.setObjectMetadata(params);  
if (result.CommonMsg.Status <= 300) {  
  console.log("Set Object(%)s's metadata successful with bucket(%s)!", params.Key, params.Bucket);  
  console.log("RequestId: %s", result.CommonMsg.RequestId);  
  return;  
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
setObjectMetadata();
```

8.2 Obtaining Object Metadata (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Object metadata contains a set of name-value pairs that are used for describing and managing objects.

Currently, only the system-defined metadata is supported. System-defined metadata consists of system-controlled metadata and user-controlled metadata. The kind of metadata like **Last-Modified** is controlled by the system and cannot be modified. However, the kind of metadata configured for objects such as **ContentLanguage** can be modified by calling APIs.

This API sends a HEAD request for obtaining object metadata.

Restrictions

- To obtain object metadata, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getObjectMetadata(params)
```

Request Parameters

Table 8-9 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

Responses

Table 8-10 Responses

Type	Description
Table 8-11 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-11 .

Table 8-11 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-12 .

Parameter	Type	Description
InterfaceResult	Table 8-13	Explanation: Results outputted for a successful call. For details, see Table 8-13 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-12 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-13 GetObjectMetadataOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Storage class configured when copying the object. If you do not specify this parameter, the object inherits the storage class of the bucket. Value range: See Table 8-14 .
AllowOrigin	string	Explanation: If Origin in the request meets the CORS rules of the bucket, AllowedOrigin specified in the CORS rules is returned. AllowedOrigin indicates the origin from which requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain at most one wildcard character (*). Example: https://*.vbs.example.com Default value: None
AllowHeader	string	Explanation: If RequestHeader in the request meets the CORS rules of the bucket, AllowedHeader specified in the CORS rules is returned. AllowedHeader indicates the allowed headers for cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Default value: None

Parameter	Type	Description
AllowMethod	string	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method allowed for cross-origin requests, that is, the operation type of buckets and objects. Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None
ExposeHeader	string	Explanation: ExposeHeader in the CORS rules of the bucket. It specifies the CORS-allowed additional headers in the response. These headers provide additional information to clients. By default, your browser can only access headers Content-Length and Content-Type. If your browser needs to access other headers, add them to a list of the allowed additional headers. Restrictions: Spaces, asterisks (*), ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Default value: None

Parameter	Type	Description
MaxAgeSeconds	number	Explanation: MaxAgeSeconds in the CORS rules of the bucket. It specifies the time your client can cache the response for a cross-origin request. Restrictions: Each CORS rule can contain at most one MaxAgeSeconds . Value range: 0 to $(2^{31} - 1)$, in seconds Default value: 100
ContentLength	number	Explanation: Object size Value range: 0 to $(2^{63} - 1)$, in bytes Default value: None
ContentType	string	Explanation: MIME type of the object. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify this parameter when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to this parameter.
LastModified	string	Explanation: Time when the object was last modified. Restrictions: The time must be in the ISO8601 format, for example, 2018-01-01T00:00:00.000Z . Default value: None

Parameter	Type	Description
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object. Value range: The value must contain 32 characters. Default value: None
VersionId	string	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
Restore	string	Explanation: Restoration status of an object. For an Archive object that is being restored or has been restored, this header is returned. For example, ongoing-request="true" indicates that the object is being restored. ongoing-request="false", expiry-date="Wed, 7 Nov 2012 00:00:00 GMT" indicates that the object has been restored, where expiry-date indicates when the restored object expires. Restrictions: If the object is not in the Archive storage class, this parameter is left blank. Default value: None

Parameter	Type	Description
Expiration	string	Explanation: Expiration details, for example, " expiry-date= \\"Mon, 11 Sep 2023 00:00:00 GMT \\" Default value: None
WebsiteRedirectLocation	string	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation . Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Default value: None

Parameter	Type	Description
Metadata	object	Explanation: Custom metadata of the object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Default value: None

Table 8-14 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

You can call **ObsClient.getObjectMetadata** to obtain the metadata of an object, including its length, MIME type, and custom metadata. The following code shows how to obtain the metadata of object **example/objectname** in bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectMetadata() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
    };
    // Obtain the object metadata.
    const result = await obsClient.getObjectMetadata(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s, ContentType:%s, ContentLength:%d, LastModified:%s\n",
        result.InterfaceResult.StorageClass, result.InterfaceResult.ETag, result.InterfaceResult.ContentType,
        result.InterfaceResult.ContentLength, result.InterfaceResult.LastModified);
      // Obtain the custom metadata of the object.
      console.log("Metadata:%v", result.InterfaceResult.Metadata);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

getObjectMetadata();
```

8.3 Configuring an Object ACL (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Access control lists (ACLs) allow resource owners to grant other accounts the permissions to access resources. By default, only the resource owner has full control over resources when a bucket or object is created. That is, the bucket creator has full control over the bucket, and the object uploader has full control over the object. Other accounts do not have the permissions to access resources. If resource owners want to grant other accounts the read and write permissions on resources, they can use ACLs. ACLs grant permissions to accounts. After an account is granted permissions, both the account and its IAM users can access the resources.

For more information, see [ACLs](#).

An object ACL can be configured in any of the following ways:

1. Specify a pre-defined ACL when uploading an object. For details, see [Code Examples: Specifying a Pre-defined ACL During Object Creation](#).
2. Call `ObsClient.setObjectAcl` to specify a pre-defined ACL. For details, see [Code Examples: Specifying a Pre-defined ACL for an Existing Object](#).
3. Call `ObsClient.setObjectAcl` to directly set an ACL. For details, see [Code Examples: Setting an ACL for an Object Directly](#).

Restrictions

- To configure an object ACL, you must be the bucket owner or have the required permission (**obs:object:PutObjectAcl** in IAM or **PutObjectAcl** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- An object ACL supports a maximum of 100 **Grants**.

Method

```
ObsClient.setObjectAcl(params)
```

Request Parameters

Table 8-15 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ACL	AclType	No	Explanation: Pre-defined ACL Restrictions: You must specify either ACL or the combination of Owner and Grants . Value range: See Table 8-16 . Default value: None
Owner	Owner	No	Explanation: Bucket owner Restrictions: • Owner and Grants must be used together, and they cannot be used with ACL. • You must specify either ACL or the combination of Owner and Grants. Value range: See Table 8-17 . Default value: None
Delivered	boolean	No	Explanation: Whether the ACL of the bucket applies to its objects Value range: • true: The ACL of the bucket applies to its objects. • false: The ACL of the bucket does not apply to its objects. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Grants	Grant[]	No	Explanation: Grantees and permissions Restrictions: • Owner and Grants must be used together, and they cannot be used with ACL. • You must specify either ACL or the combination of Owner and Grants. Value range: See Table 8-18 . Default value: None

Table 8-16 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 8-17 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 8-18 Grant

Parameter	Type	Mandatory (Yes/No)	Description
Grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 8-19 .
Permission	PermissionType	Yes if used as a request parameter	Explanation: Granted permission Value range: See Table 8-22 . Default value: None

Table 8-19 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
Type	string	Yes if used as a request parameter	Explanation: Grantee type Value range: See Table 8-20 . Default value: None
ID	string	Yes if this parameter is used as a request parameter and Type is set to a user	Explanation: Account (domain) ID of the grantee. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Name	string	No if used as a request parameter	Explanation: Account name of the grantee. Restrictions: • The account name cannot contain full-width characters. • The account name starts with a letter. • The account name contains 6 to 32 characters. • The account name contains only letters, digits, hyphens (-), or underscores (_). Default value: None
URI	Group UriType	Yes if this parameter is used as a request parameter and Type is set to a group	Explanation: Authorized user group Value range: See Table 8-21 . Default value: None

Table 8-20 GranteeType

Constant	Description
Group	Grants permissions to user groups.
CanonicalUser	Grants permissions to individual users.

Table 8-21 GroupUriType

Constant	Default Value	Description
ObsClient.enums.GroupAllUsers	AllUsers	All users.
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.
ObsClient.enums.GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 8-22 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadACP	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.

Constant	Default Value	Description
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Responses

Table 8-23 Responses

Type	Description
Table 8-24 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-24 .

Table 8-24 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-25 .
InterfaceResult	Table 8-26	Explanation: Results outputted for a successful call. For details, see Table 8-26 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-25 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-26 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples: Specifying a Pre-defined ACL During Object Creation

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an ObsClient instance.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/
```

```
intl/en-us/usermanual-ca/ca_01_0003.html.  
access_key_id: process.env.ACCESS_KEY_ID,  
secret_access_key: process.env.SECRET_ACCESS_KEY,  
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
variables or import an AK/SK pair in other ways.  
// security_token: process.env.SECURITY_TOKEN,  
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here  
as an example. Replace it with the one currently in use.  
server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function putObject() {  
  try {  
    const params = {  
      // Specify the bucket name.  
      Bucket: "examplebucket",  
      // Specify the object. example/objectname is used in this example.  
      Key: "example/objectname",  
      // Specify a text object.  
      Body: 'Hello OBS',  
      // Set the object ACL to public read.  
      ACL: obsClient.enums.AclPublicRead  
    };  
    // Upload the object.  
    const result = await obsClient.putObject(params);  
    if (result.CommonMsg.Status <= 300) {  
      console.log("Put object(%s) under the bucket(%s) successful!!", params.Key, params.Bucket);  
      console.log("RequestId: %s", result.CommonMsg.RequestId);  
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);  
      return;  
    }  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  }  
};  
putObject();
```

Code Examples: Specifying a Pre-defined ACL for an Existing Object

Sample code:

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
  coding may result in leakage.  
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
  variables or import an AK/SK pair in other ways.  
  // security_token: process.env.SECURITY_TOKEN,
```

```
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Set the object ACL to private read and write.
      ACL: obsClient.enums.AclPrivate
    };
    // Set the object ACL.
    const result = await obsClient.setObjectAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set Object(%s)'s acl successful with Bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

setObjectAcl();
```

Code Examples: Setting an ACL for an Object Directly

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
```

```
// Specify an object. example/objectname is used in this example.
Key: "example/objectname",
// Specify the version ID of the object.
VersionId: 'G001117FCE89978B0000401205D5DC9A',
// Specify the owner of the object.
Owner: { 'ID': 'ownerid' },
// Specify the information about the authorized user.
Grants: [
  // Grant the write permission to a specified user (0a03f5833900d3730f13c00f49d5exxx in this
  // example).
  { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
  obsClient.enums.PermissionWrite },
  // Grant the read permission to all users.
  { Grantee: { Type: 'Group', URI: obsClient.enums.GroupAllUsers }, Permission:
  obsClient.enums.PermissionRead },
]
};
// Set the ACL.
const result = await obsClient.setObjectAcl(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Set Object(%)s's acl successful with Bucket(%)!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
  attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
}
};
setObjectAcl();
```

8.4 Obtaining the ACL of an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Access control lists (ACLs) allow resource owners to grant other accounts the permissions to access resources. By default, only the resource owner has full control over resources when a bucket or object is created. That is, the bucket creator has full control over the bucket, and the object uploader has full control over the object. Other accounts do not have the permissions to access resources. If resource owners want to grant other accounts the read and write permissions on resources, they can use ACLs. ACLs grant permissions to accounts. After an account is granted permissions, both the account and its IAM users can access the resources.

For more information, see [ACLs](#).

This API returns the ACL of an object.

Restrictions

- To obtain an object ACL, you must be the bucket owner or have the required permission (**obs:object:GetObjectAcl** in IAM or **GetObjectAcl** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getObjectAcl(params)
```

Request Parameters

Table 8-27 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None

Responses

Table 8-28 Responses

Type	Description
Table 8-29 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-29 .

Table 8-29 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-30 .
InterfaceResult	Table 8-31	Explanation: Results outputted for a successful call. For details, see Table 8-31 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-30 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-31 GetObjectAclOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
VersionId	string	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A
Owner	Owner	Explanation: Object owner. For details, see Owner .
Delivered	string	Explanation: Whether the bucket ACL is applied to the objects in the bucket.
Grants	Grant[]	Explanation: Grantees' permission information. For details, see Table 8-33 .

Table 8-32 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 8-33 Grant

Parameter	Type	Mandatory (Yes/No)	Description
Grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 8-34 .
Permission	PermissionType	Yes if used as a request parameter	Explanation: Granted permission Value range: See Table 8-37 .

Table 8-34 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
Type	string	Yes if used as a request parameter	Explanation: Grantee type Value range: See Table 8-35 . Default value: None
ID	string	Yes if this parameter is used as a request parameter and Type is set to a user	Explanation: Account (domain) ID of the grantee. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Name	string	No if used as a request parameter	Explanation: Account name of the grantee. Restrictions: • The account name cannot contain full-width characters. • The account name starts with a letter. • The account name contains 6 to 32 characters. • The account name contains only letters, digits, hyphens (-), or underscores (_). Default value: None
URI	Group UriType	Yes if this parameter is used as a request parameter and Type is set to a group	Explanation: Authorized user group Value range: See Table 8-36 . Default value: None

Table 8-35 GranteeType

Constant	Description
Group	Grants permissions to user groups.
CanonicalUser	Grants permissions to individual users.

Table 8-36 GroupUriType

Constant	Default Value	Description
ObsClient.enums.GroupAllUsers	AllUsers	All users.
ObsClient.enums.GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.
ObsClient.enums.GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 8-37 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadAcp	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Code Examples

This example returns the ACL of object **example/objectname**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
```

```
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
    };
    // Obtain the object ACL.
    const result = await obsClient.getObjectAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s)'s acl successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Owner[ID]: %s', result.InterfaceResult.Owner.ID);
      console.log('Owner[Name]: %s', result.InterfaceResult.Owner.Name);
      for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
        const grant = result.InterfaceResult.Grants[i];
        fmt.Printf("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s\n",
          index, grant.GrantType, grant.Grantee.ID, grant.Grantee.URI, grant.Permission)
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

getObjectAcl();
```

8.5 Listing Objects in a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API lists some or all of the objects in a bucket. You can configure the prefix, number, or start position as the filter to list objects. Returned objects are listed in alphabetical order by object name.

Restrictions

- A maximum of 1,000 objects can be listed for each API call.
- To list objects in a bucket, you must be the bucket owner or have the required permission (**obs:bucket:ListBucket** in IAM or **ListBucket** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.listObjects(params)
```

Request Parameters

Table 8-38 ListObjectsInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Marker	string	No	Explanation: Object name to start with when listing objects in a bucket. Objects after the specified object are returned in alphabetical order. Restrictions: This parameter is only used for listing objects with a single version. Value range: The value can contain 1 to 1,024 characters. Default value: None
Prefix	string	No	Explanation: Name prefix that the objects to be listed must contain Assume that you have the following objects: logs/day1 , logs/day2 , logs/day3 , and ExampleObject.jpg . If you specify logs/ as the prefix, logs/day1 , logs/day2 , and logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all objects in the bucket will be returned. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
MaxKeys	number	No	Explanation: The maximum number of objects returned in the response in alphabetical order Restrictions: None Value range: The value ranges from 1 to 1000 . If the specified value is greater than 1000 , only 1,000 objects are returned. Default value: 1000

Parameter	Type	Mandatory (Yes/No)	Description
Delimiter	string	No	Explanation: Object names are grouped by this parameter, which is often used with Prefix. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes. Assume that a bucket has objects abcd, abcde, and bbcde in it. If Delimiter is set to d and Prefix is set to a, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcde as the prefix. For a parallel file system, if this parameter is not specified, all the content in the directory is recursively listed by default, and subdirectories are also listed. In big data scenarios, parallel file systems usually have multiple directory levels, and each directory level has a large number of objects. In such case, you are advised to configure [delimiter=/] to list only the content in the current directory, without the content in subdirectories, to make the listing more efficient. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Encoding Type	string	No	Explanation: Encoding type for some elements in the response. If Delimiter , Marker , Prefix , CommonPrefixes , NextMarker , and Key contain control characters (special characters) that are not supported by the XML 1.0 standard, set this parameter to url . Restrictions: None Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Responses

Table 8-39 Responses

Type	Description
Table 8-40 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-40 .

Table 8-40 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-41 .

Parameter	Type	Description
InterfaceResult	Table 8-42	Explanation: Results outputted for a successful call. For details, see Table 8-42 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-41 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-42 ListObjectsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
Location	string	Explanation: Region where a bucket is located.
Bucket	string	Explanation: Bucket name
Prefix	string	Explanation: Object name prefix, which is consistent with that set in the request. Assume that you have the following objects: logs/day1 , logs/day2 , logs/day3 , and ExampleObject.jpg . If you specify logs/ as the prefix, logs/day1 , logs/day2 , and logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all objects in the bucket will be returned.
Marker	string	Explanation: Object name to start with when listing objects. Objects after the specified object are returned in alphabetical order.

Parameter	Type	Description
Delimiter	string	Explanation: Object names are grouped by this parameter, which is often used with Prefix. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes. Assume that a bucket has objects abcd, abcde, and bbcde in it. If Delimiter is set to d and Prefix is set to a, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcd as the prefix. For a parallel file system, if this parameter is not specified, all the content in the directory is recursively listed by default, and subdirectories are also listed. In big data scenarios, parallel file systems usually have multiple directory levels, and each directory level has a large number of objects. In such case, you are advised to configure [delimiter=/] to list only the content in the current directory, without the content in subdirectories, to make the listing more efficient.
MaxKeys	number	Explanation: The maximum number of objects returned in the response in alphabetical order. This parameter corresponds to the same parameter in the request.

Parameter	Type	Description
IsTruncated	string	Explanation: Whether all results are returned in the response. A maximum of 1,000 objects can be listed at a time. If the number of objects is greater than 1,000, the objects beyond 1,000 cannot be returned. Value range: • true: Not all results are returned. • false: All results are returned.
NextMarker	string	Explanation: Where in the bucket the next listing begins. If not all results are returned, the response contains this parameter to mark the last object listed in the request. In a subsequent request, you can set Marker to the value of this parameter to list the remaining objects.
Contents	Content[]	Explanation: List of objects in the bucket. For details, see Content .
CommonPrefixes	CommonPrefix[]	Explanation: List of object name prefixes grouped based on the Delimiter parameter (if specified) Value range: See Table 8-44 .
EncodingType	string	Explanation: Encoding type for some elements in the response. If Delimiter , Marker , Prefix , CommonPrefixes , NextMarker , and Key contain control characters (special characters) that are not supported by the XML 1.0 standard, set this parameter to url .

Table 8-43 Content

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
LastModified	string	Explanation: Time when the object was last modified, in UTC.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. ETag is the unique identifier of the object contents and is used to determine whether the contents of an object are changed. For example, if the ETag value is A when an object is uploaded and B when the object is downloaded, this indicates the contents of the object have been changed. The ETag reflects changes to an object's contents, not its metadata. Objects created by the upload and copy operations are assigned unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
Size	number	Explanation: Object size, in bytes.
Owner	Owner	Explanation: Object owner. The value contains the domain ID and name of the object owner. For details, see Table 8-45 .
StorageClass	string	Explanation: Object storage class.
Type	string	Explanation: Type of the object Value range: • NORMAL indicates a regular object. • APPENDABLE indicates an appendable object.

Table 8-44 CommonPrefix

Parameter	Type	Description
Prefix	string	Explanation: Prefix of objects in a bucket Value range: The value can contain 1 to 1,024 characters. Default value: None

Table 8-45 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Code Examples: Simple Listing

The following sample code shows how to list objects in simple mode. A maximum of 1,000 objects can be returned.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
```



```
variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // List objects in a bucket.
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listObjects();
```

NOTE

- A maximum of 1,000 objects can be listed each time. If a bucket contains more than 1,000 objects, **InterfaceResult.IsTruncated** in the response is **true**, indicating not all objects were listed. In such case, you can use **InterfaceResult.NextMarker** to obtain the start position for next listing.
- If you want to obtain all objects in a specified bucket, use pagination. For details, see [Code Examples: Listing All Objects Using Pagination](#).

Code Examples: Listing a Specified Number of Objects

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  access_key_id: process.env.ACCESS_KEY_ID,
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
```

```
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is 1000. 100 is used in this example.
      MaxKeys: 100,
    };
    // List objects in a bucket.
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

listObjects();
```

Code Examples: Listing Objects by Prefix

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
async function listObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will
      // be listed.
      Prefix: "test/",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is
      // 1000. 100 is used in this example.
      MaxKeys: 100,
    };

    // List objects in a bucket.
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listObjects();
```

Code Examples: Specifying a Start Position for Listing

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
```

```
try {
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket",
    // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will
    be listed.
    Prefix: "test/",
    // Specify the maximum number of objects to be returned in alphabetic order. The default value is
    1000. 100 is used in this example.
    MaxKeys: 100,
    // Specify the position (test/test2 in this example) where the object listing starts.
    Marker: "test/test2",
  };
  // List objects in a bucket.
  const result = await obsClient.listObjects(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("List objects under the bucket(%s) successful!", params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
      const val = result.InterfaceResult.Contents[j];
      console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
        j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
    };
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
  response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
  attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

listObjects();
```

Code Examples: Listing All Objects Using Pagination

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
```

```
try {
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket",
    // Specify an encoding type. url is used in this example. If the objects to list contain special characters,
    // this parameter must be passed.
    EncodingType: "url",
  };
  while (true) {
    // List object versions in the bucket.
    const result = await obsClient.listObjects(params);
    if (result.CommonMsg.Status > 300) {
      console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
      console.log("Status: %d", result.CommonMsg.Status);
      console.log("Code: %s", result.CommonMsg.Code);
      console.log("Message: %s", result.CommonMsg.Message);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("List objects under the bucket(%s) successful!", params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
      const val = result.InterfaceResult.Contents[j];
      console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
        j, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
    }
    if (result.InterfaceResult.IsTruncated === "true") {
      params.KeyMarker = result.InterfaceResult.NextKeyMarker;
    } else {
      break;
    }
  }
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
}
};
listObjects();
```

Code Examples: Listing All Objects in a Folder

OBS does not involve folders like in a file system. All elements in buckets are objects. Folders are actually objects whose sizes are 0 and whose names end with a slash (/). When you set a folder name as the prefix, objects in this folder will be listed. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
```

```
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will
      // be listed.
      Prefix: "test/",
      // Specify an encoding type. url is used in this example. If the objects to list contain special characters,
      // this parameter must be passed.
      EncodingType: "url",
    };
    let count = 1
    while (true) {
      // List object versions in the bucket.
      const result = await obsClient.listObjects(params);
      if (result.CommonMsg.Status > 300) {
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
        return;
      }
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        const val = result.InterfaceResult.Contents[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, LastModified:%s, Size:%d',
          count++, val.Owner.ID, val.ETag, val.Key, val.LastModified, val.Size);
      }
      if (result.InterfaceResult.IsTruncated === "true") {
        params.Marker = result.InterfaceResult.NextMarker;
      } else {
        break;
      }
    }
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

listObjects();
```

8.6 Deleting an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

NOTE

Exercise caution when performing this operation. If the versioning function is disabled for the bucket where the object is located, the object cannot be restored after being deleted.

Function

This API deletes an object in the specified bucket to save space and costs.

Restrictions

- To delete an object, you must be the bucket owner or have the required permission (**obs:object:DeleteObject** in IAM or **DeleteObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- If versioning is not enabled for a bucket, deleted objects cannot be restored.

Method

```
ObsClient.deleteObject(params)
```

Request Parameters

Table 8-46 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None

Responses

Table 8-47 Responses

Type	Description
Table 8-48 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-48 .

Table 8-48 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-49 .
InterfaceResult	Table 8-50	Explanation: Results outputted for a successful call. For details, see Table 8-50 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-49 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-50 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
DeleteMarker	boolean	Explanation: Whether the deleted object is a delete marker. Value range: • true: The deleted object is a delete marker. • false: The deleted object is not a delete marker.
VersionId	string	Explanation: ID of the object version to be deleted, for example, G001117FCE89978B0000401205D5DC9

Code Examples

This example deletes object **example/objectname** from bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
```

```
in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example) to delete.
      Key: 'example/objectname',
    };
    // Delete the object.
    const result = await obsClient.deleteObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

deleteObject();
```

8.7 Batch Deleting Objects (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

NOTE

Exercise caution when performing this operation. If versioning is disabled for the bucket where objects are located, objects cannot be restored after being deleted.

Function

This API deletes objects in a batch from a specific bucket. Deleted objects cannot be restored.

In a batch deletion, OBS concurrently deletes the specified objects and returns the deletion result of each object.

Restrictions

- To delete objects in a batch, you must be the bucket owner or have the required permission (**obs:object:DeleteObject** in IAM or **DeleteObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- If versioning is not enabled for a bucket, deleted objects cannot be restored.
- A maximum of 1,000 objects can be deleted at a time. If you send a request for deleting more than 1,000 objects, OBS returns an error message.
- After concurrent tasks are assigned, if an internal error occurs during cyclic deletion of multiple objects, an object may be deleted in the index data but still exist in the metadata.

Method

```
ObsClient.deleteObjects(params)
```

Request Parameters

Table 8-51 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Quiet	boolean	No	Explanation: Mode of the response to the request for deleting objects in a batch Restrictions: None Value range: false: The detailed mode. Results of both successful and failed deletions are returned. true: The quiet mode. Only results of failed deletions are returned. Default value: false
Objects	ObjectToDelete[]	Yes	Explanation: List of objects to be deleted. For details, see Table 8-52 . Restrictions: A maximum of 1000 objects can be deleted at a time. Value range: None Default value: None
EncodingType	string	No	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Restrictions: None Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Table 8-52 ObjectToDelete

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: ID of the object version to be deleted, for example, G001117FCE89978B0000401205D5DC9 . Restrictions: None Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is deleted.

Responses

Table 8-53 Responses

Type	Description
Table 8-54 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-54 .

Table 8-54 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-55 .
InterfaceResult	Table 8-56	Explanation: Results outputted for a successful call. For details, see Table 8-56 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-55 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-56 DeleteObjectsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
DeletedIds	Deleted[]	Explanation: List of objects that were successfully deleted. For details, see Table 8-57 .
Errors	Error[]	Explanation: List of objects that failed to be deleted. For details, see Table 8-58 .
EncodingType	string	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key .

Table 8-57 Deleted

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .

Parameter	Type	Description
VersionId	string	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 .
DeleteMarker	bool	Explanation: Whether the deleted object is a delete marker. Value range: • true: The deleted object is a delete marker. • false: The deleted object is not a delete marker.
DeleteMarkerVersionId	string	Explanation: Version ID of a delete marker to create or delete. OBS returns this element in the response when a delete marker is created or deleted for a versioning-enabled bucket. This element will be returned in either of the following cases: • You send a delete request specifying an object's name without providing a version ID. In this case, OBS creates a delete marker and returns its version ID in the response. • You send a delete request specifying both an object's name and its version ID, but this version ID points to a delete marker. In this case, OBS deletes the delete marker and returns its version ID in the response.

Table 8-58 Error

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
VersionId	string	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 .
Code	string	Explanation: Error code for the failed deletion. For details, see OBS Error Codes .
Message	string	Explanation: Error message for the failed deletion. For details, see OBS Error Codes .

Code Examples

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObjects() {
  try {
    const params = {
      // Specify the bucket name.
```

```
Bucket: "examplebucket",
// Specify the object list to delete.
Objects: [
  { Key: 'objectname1', VersionId: "version1" },
  { Key: 'objectname2', VersionId: "version2" },
  { Key: 'objectname3', VersionId: "version3" }
]
};
// Delete the objects in a batch.
const result = await obsClient.deleteObjects(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Delete objects under the bucket(%s) successful!", params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  // Return details about which objects were deleted.
  console.log('Deleted:');
  for (let i = 0; i < result.InterfaceResult.Deleted.length; i++) {
    const deleted = result.InterfaceResult.Deleted[i];
    console.log("Deleted[%d]-Key:%s, VersionId:%s", i, deleted.Key, deleted.VersionId);
  };
  // Return details about which objects were not deleted.
  console.log('Errors:');
  for (let i = 0; i < result.InterfaceResult.Errors.length; i++) {
    const err = result.InterfaceResult.Errors[i];
    console.log("Errors[%d]-Key:%s, Code:%s", i, err.Key, err.Code);
  };
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
deleteObjects();
```

8.8 Copying an Object (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API copies an object stored in OBS to a specified path. You can copy an object of up to 5 GB in a single operation.

Restrictions

- To copy an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The target object size ranges from 0 to 5 GB. If the source object size exceeds 5 GB, you must use a multipart copying API by referring to [Copying a Part \(SDK for Node.js\)](#).

Method

```
ObsClient.copyObject(params)
```

Request Parameters

Table 8-59 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters.

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
CopySource	string	Yes	Explanation: Names of the source bucket and object. If the source object has multiple versions, the versionId parameter is used to specify the desired version. Example: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A Restrictions: None Restrictions: Full-width characters and percent signs (%) must be URL-encoded. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ACL	AclType	No	Explanation: ACL that can be pre-defined during the object copy. For details about the ACL, see ACLs . Restrictions: None Value range: See Table 8-60 . Default value: None
MetadataDirective	MetadataDirectiveType	No	Explanation: Policy for copying the source object's attributes Restrictions: None Value range: See Table 8-61 . Default value: None
CopySourceIfMatch	string	No	Explanation: If the ETag of the source object is the same as the one specified by this parameter, it is copied. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceIfNone-Match	string	No	Explanation: If the ETag of the source object is different from the one specified by this parameter, it is copied. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None
CopySourceIfUnmodifiedSince	string	No	Explanation: If the source object has not been modified since the specified time, it is copied. Otherwise, an exception is thrown. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt . Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceIfModifiedSince	string	No	Explanation: If the source object has been modified since the specified time, it is copied. Otherwise, an exception is thrown. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
CopySourceSseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceSseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
CacheControl	string	No	Explanation: Cache-Control in the response is rewritten. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ContentDisposition	string	No	Explanation: Content-Disposition in the response is rewritten. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentEncoding	string	No	Explanation: Content-Encoding in the response is rewritten. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ContentLanguage	string	No	Explanation: Content-Language in the response is rewritten. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ContentType	string	No	Explanation: Content-Type in the response is rewritten. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	string	No	Explanation: Expires in the response is rewritten. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
StorageClass	StorageClassType	No	Explanation: Storage class configured when copying the object. Restrictions: None Value range: See Table 8-62 . Default value: None. If this parameter is not specified, the storage class of the bucket is used as that of the object.

Parameter	Type	Mandatory (Yes/No)	Description
Metadata	object	No	Explanation: Custom metadata of the target object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the object to copy and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the object to copy. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the object to copy. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the object to copy, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SuccessActionRedirect	string	No	Explanation: Address (URL) to which a successfully answered request is redirected. If the value is valid and the request is successful, OBS returns status code 303. Location contains the value of this parameter, as well as the bucket name, object name, and object ETag. If the value is invalid, OBS ignores this parameter. In such case, the Location header is the object address, and OBS returns a status code based on whether the operation succeeded or failed. Restrictions: None Value range: None Default value: None

Table 8-60 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 8-61 MetadataDirectiveType

Constant	Default Value	Description
ObsClient.enums.CopyMetadata	COPY	When copying an object, the object's attributes are also copied. NOTICE This value is used only in the API for Copying an Object (SDK for Node.js) .
ObsClient.enums.ReplaceMetadata	REPLACE	REPLACE uses the complete header carried in the current request to replace the original one and deletes the metadata that is not specified.
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	REPLACE_NEW replaces the metadata that already has a value, assigns a value to the metadata that does not have a value, and retains the metadata that is not specified. NOTICE This value is used only in the API for Configuring Object Metadata (SDK for Node.js) .

Table 8-62 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

 NOTE

- If **CopySourceIfUnmodifiedSince**, **CopySourceIfMatch**, **CopySourceIfModifiedSince**, or **CopySourceIfNoneMatch** is included but the specified condition is not met, **412 precondition failed** will be returned.
- **CopySourceIfModifiedSince** and **CopySourceIfNoneMatch** can be used together. So can **CopySourceIfUnmodifiedSince** and **CopySourceIfMatch**.

Responses

Table 8-63 Responses

Type	Description
Table 8-64 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 8-64 .

Table 8-64 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 8-65 .
InterfaceResult	Table 8-66	Explanation: Results outputted for a successful call. For details, see Table 8-66 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 8-65 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .

Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 8-66 CopyObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
LastModified	string	Explanation: Time when the target object was last modified, in UTC
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of the target object. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and B when the object is downloaded, this indicates the contents of the object have been changed. The ETag reflects changes to an object's contents, not its metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.

Parameter	Type	Description
CopySourceVersionId	string	Explanation: Version ID of the source object
VersionId	string	Explanation: Version ID of the target object

Code Examples: Simple Copying

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // Specify the target bucket name.
      Bucket: "examplebucket",
      // Specify the name of the object copy (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the names of the source bucket (sourcebucketname in this example) and source object
      // (sourceobjectkey in this example).
      CopySource: 'sourcebucketname/sourceobjectname'
    };
    // Copy the source object.
    const result = await obsClient.copyObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
        params.Key, params.CopySource);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
        result.InterfaceResult.LastModified);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
      response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
```

```
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
copyObject();
```

Code Examples: Rewriting Object Attributes

NOTE

Use the **Metadata** parameter to specify the object's customized metadata to be rewritten and the **MetadataDirective** parameter to specify the rewrite mode, which can be **ObsClient.enums.ReplaceMetadata** (rewrite) or **ObsClient.enums.CopyMetadata** (copy from the source object).

The following sample code shows how to rewrite object properties.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // Specify the target bucket name.
      Bucket: "examplebucket",
      // Specify the name of the object copy (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the names of the source bucket (sourcebucketname in this example) and source object
      // (sourceobjectkey in this example).
      CopySource: 'sourcebucketname/sourceobjectname'
    };
    // Specify custom metadata.
    Metadata: {'property':'property-value'},
    // Specify the policy for copying the metadata of the source object. In this example,
    // ObsClient.enums.ReplaceMetadata is used to rewrite the metadata of the source object.
    MetadataDirective: ObsClient.enums.ReplaceMetadata
  };
  // Copy the object.
  const result = await obsClient.copyObject(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
      params.Key, params.CopySource);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
      result.InterfaceResult.LastModified);
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
copyObject();
```

Code Examples: Specifying Copying Conditions

When copying an object, you can specify one or more restriction conditions. If the conditions are met, the object will be copied. Otherwise, an error code will be returned and the copy will fail.

NOTE

- The ETag of the source object is the MD5 check value of the source object.
- If the object copy request includes **CopySourceIfUnmodifiedSince**, **CopySourceIfMatch**, **CopySourceIfModifiedSince**, or **CopySourceIfNoneMatch**, and the specified condition is not met, the object copy will fail with error code **412 Precondition Failed** returned.
- **CopySourceIfModifiedSince** and **CopySourceIfNoneMatch** can be used together. So do **CopySourceIfUnmodifiedSince** and **CopySourceIfMatch**.

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the name of the object copy (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the names of the source bucket (sourcebucketname in this example) and source object (sourceobjectkey in this example).
      CopySource: 'sourcebucketname/sourceobjectname'
    };
    // Specify the copy conditions. If the source object has been modified since the specified time, it will be copied.
```

```
CopySourceIfModifiedSince: 'Thu, 31 Dec 2015 16:00:00 GMT',
// Specify the copy conditions. If the ETag of the source object is different from the one specified by this
parameter, the source object will be copied.
CopySourceIfNoneMatch: 'none-match-etag'
};
// Copy the object.
const result = await obsClient.copyObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
params.Key, params.CopySource);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
result.InterfaceResult.LastModified);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyObject();
```

Code Examples: Rewriting an Object ACL

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the name of the object copy (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the names of the source bucket (sourcebucketname in this example) and source object
(sourceobjectkey in this example).
      CopySource: 'sourcebucketname/sourceobjectname'
      // Rewrite the object ACL to public read during the copy.
      ACL : obsClient.enums.AclPublicRead
    };
  }
}
```

```
};  
// Copy the object.  
const result = await obsClient.copyObject(params);  
if (result.CommonMsg.Status <= 300) {  
  console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,  
params.Key, params.CopySource);  
  console.log("RequestId: %s", result.CommonMsg.RequestId);  
  console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,  
result.InterfaceResult.LastModified);  
  return;  
};  
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
console.log("Status: %d", result.CommonMsg.Status);  
console.log("Code: %s", result.CommonMsg.Code);  
console.log("Message: %s", result.CommonMsg.Message);  
console.log("RequestId: %s", result.CommonMsg.RequestId);  
} catch (error) {  
  console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
  console.log(error);  
};  
};  
copyObject();
```

9 Multipart Upload (SDK for Node.js)

9.1 Multipart Upload APIs (SDK for Node.js)

To upload a large file, multipart upload is recommended. Multipart upload is applicable to many scenarios. Below are some examples.

- Files to be uploaded are larger than 100 MB.
- The network connection to the OBS server breaks often.
- Sizes of files to be uploaded are uncertain.

Multipart upload consists of three phases:

1. [Initiating a Multipart Upload \(SDK for Node.js\)](#)
2. [Uploading a Part \(SDK for Node.js\)](#)
3. [Assembling Parts \(SDK for Node.js\)](#) or [Aborting a Multipart Upload \(SDK for Node.js\)](#)

Multipart upload is mainly used for large file upload or when the network connection is poor. This example uploads a large file in parts concurrently:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require('esdk-obs-nodejs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
const fs = require('fs');

// Create an ObsClient instance.
var obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
```

```
});

// Specify the bucket name.
const Bucket = 'bucketname'
// Specify an object (example/objectname in this example).
const Key = 'objectname'

// Specify the part size.
const DEFAULT_PART_SIZE = 9 * 1024 * 1024;
// Set the number of concurrent parts to 20.
const CONCURRENCY = 20

// Prepare multipart upload parameters.
const preparePartParams = (Bucket, Key, UploadId) => (sampleFile, partSize = DEFAULT_PART_SIZE) => {
  try {
    const fileSize = fs.lstatSync(sampleFile).size;
    const partCount = fileSize % partSize === 0 ? Math.floor(fileSize / partSize) : Math.floor(fileSize /
partSize) + 1;

    const uploadPartParams = [];
    // Specify the concurrent upload.
    for (let i = 0; i < partCount; i++) {
      // Start position of parts in the file
      let Offset = i * partSize;
      // Part size
      let currPartSize = (i + 1 === partCount) ? fileSize - Offset : partSize;
      // Part number
      let PartNumber = i + 1;
      uploadPartParams.push({
        Bucket,
        Key,
        PartNumber,
        UploadId,
        Offset,
        SourceFile: sampleFile,
        PartSize: currPartSize,
      });
    };

    return ({ uploadPartParams, fileSize });
  } catch (error) {
    console.log(error)
  };
};

/**
 * uploadSuccessSize: Size of the parts that have been uploaded
 * uploadSuccessCount: Number of the parts that have been uploaded
 * concurrency: Current concurrency
 */
let uploadSuccessSize = 0;
let uploadSuccessCount = 0;
let concurrency = 0

const parts = [];

// Upload parts.
const uploadPart = (uploadPartParam, otherUploadPartInfo) => {
  const partCount = otherUploadPartInfo.partCount;
  const fileSize = otherUploadPartInfo.fileSize;
  concurrency++;
  return obsClient
    .uploadPart(uploadPartParam)
    .then(result => {
      const { PartNumber, PartSize } = uploadPartParam;
      if (result.CommonMsg.Status < 300) {
        uploadSuccessCount++;
        uploadSuccessSize += PartSize;
      }
    })
};
```

```
        // Print the upload progress.
        console.log(`the current concurrent count is ${concurrency} | uploaded segment: $
{uploadSuccessCount}/${partCount}. the progress is $${(uploadSuccessSize / fileSize) * 100}.toFixed(2)}% |
the partNumber ${PartNumber} upload succeeded.`);
        parts.push({ PartNumber, ETag: result.InterfaceResult.ETag });
    } else {
        console.log(result.CommonMsg.Code, parts);
    };
    concurrency--;
}).catch(function (err) {
    console.log(err);
    throw err;
});

// Multipart upload
const uploadFile = (sourceFile) => {
    // Initiate the multipart upload task.
    obsClient.initiateMultipartUpload({
        Bucket,
        Key
    }).then(res => {
        const Status = res.CommonMsg.Status;
        const UploadId = res.InterfaceResult.UploadId;

        if (typeof Status === 'number' && Status > 300) {
            console.log(`initiateMultipartUpload failed! Status:${Status}`);
            return;
        };

        const partParams = preparePartParams(Bucket, Key, UploadId)(sourceFile);
        const uploadPartParams = partParams.uploadPartParams;
        const fileSize = partParams.fileSize;
        const partCount = uploadPartParams.length;
        const otherUploadPartInfo = { fileSize, partCount };

        // Call the parallel upload function.
        parallelFunc(uploadPartParams, (param) => uploadPart(param, otherUploadPartInfo),
CONCURRENCY)
            .then(() => {
                // Assemble parts.
                obsClient.completeMultipartUpload({
                    Bucket,
                    Key,
                    UploadId,
                    Parts: parts.sort((a, b) => a.PartNumber - b.PartNumber)
                }, (err, result) => {
                    if (err) {
                        console.log('Error-->' + err);
                    } else {
                        console.log('Status-->' + result.CommonMsg.Status);
                    };
                });
            });

    }).catch(function (err) {
        console.log(err)
    });
};

/**
 * Implement the parallel function execution.
 * @param {Array} params Parameter array of the callback function
 * @param {Promise} promiseFn Callback function
 * @param {number} limit Number of parallel parts
 */
const parallelFunc = (params, promiseFn, limit) => {
    return new Promise((resolve) => {
```



```
let concurrency = 0;
let finished = 0;
const count = params.length;
const run = (param) => {
  concurrency++;
  promiseFn(param)
    .then(() => {
      concurrency--;
      drainQueue();
      finished++;
      if (finished === count) {
        resolve();
      }
    });
};
const drainQueue = () => {
  while (params.length > 0 && concurrency < limit) {
    var param = params.shift();
    run(param);
  }
};
drainQueue();
});

uploadFile('localfile');
```

NOTE

When uploading a large file in parts, you need to use the **Offset** and **PartSize** parameters to specify the start and end positions of each part in the file.

CAUTION

If the concurrency value is too great, timeout may occur due to network instability. In such case, you need to reduce that value.

9.2 Initiating a Multipart Upload (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API initiates a multipart upload and returns a globally unique upload ID. You can use this upload ID in your subsequent requests for uploading, assembling, and listing parts. Initiating multipart upload tasks do not affect existing objects with the same names as the involved objects in those tasks. There can be more than one multipart upload for the same object. Each multipart upload initiation request can contain additional headers such as **contentType**, **contentEncoding**, and the headers for ACL and custom metadata. These headers are recorded in the multipart upload metadata. After the API for initiating a multipart upload is

successfully called, an upload ID will be returned. This ID will be used in subsequent operations.

Restrictions

- To initiate a multipart upload, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- After a multipart upload was initiated and one or more parts have been uploaded, you will be charged for storing these parts until you complete or cancel the upload.

Method

```
ObsClient.initiateMultipartUpload(params)
```

Request Parameters

Table 9-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
ACL	AclType	No	Explanation: An access control list (ACL) that can be pre-defined when creating a bucket. For details about ACLs, see ACLs . Restrictions: None Value range: See Table 9-2 . Default value: private

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation. Value range: None Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentType	string	No	Explanation: MIME type of the file to be uploaded. MIME type is a standard way of describing a data type and is used by the browser to decide how to display data. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: If you do not specify this parameter when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to this parameter.
StorageClass	StorageClassType	No	Explanation: Object storage class. If this parameter is not specified, the object inherits the storage class of its bucket. Restrictions: None Value range: See Table 9-3. Default value: STANDARD

Parameter	Type	Mandatory (Yes/No)	Description
Metadata	object	No	Explanation: Custom metadata of the object to be uploaded. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the current object and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the current object. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the current object, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	number	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: - The value cannot be smaller than the number of days that have passed since the object was created. For example, if the object was uploaded 10 days ago, you cannot specify a value less than 10. - This parameter can only be configured when uploading an object. Value range: 1 to ($2^{63} - 1$), in days Default value: None
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: • <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. • <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). • <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: • If this parameter is not specified, the default master key will be used. • If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
EncodingType	string	No	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Table 9-2 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 9-3 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 9-4 Responses

Type	Description
Table 9-5 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-5 .

Table 9-5 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-6 .

Parameter	Type	Description
InterfaceResult	Table 9-7	Explanation: Results outputted for a successful call. For details, see Table 9-7 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-6 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-7 InitiateMultipartUploadOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
Bucket	string	Explanation: Name of the bucket involved in the multipart upload.
Key	string	Explanation: Object name in the multipart upload. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
UploadId	string	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 .
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Parameter	Type	Description
EncodingType	string	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Code Examples

This example initiates a multipart upload for object **example/objectname**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function initiateMultipartUpload() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: "example/objectname",
    };
    // Initiate the multipart upload.
    const result = await obsClient.initiateMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Initiate multipart upload successfully with bucket(%s) and object(%s)!", params.Bucket,
        params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // A globally unique identifier of the multipart upload task, which will be used in uploading and
      // assembling parts.
      console.log("UploadId: %s", result.InterfaceResult.UploadId);
      return;
    }
  }
}
```

```
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
initiateMultipartUpload();
```

Helpful Links

- [Uploading a Part](#)
- [OBS Error Codes](#)

9.3 Uploading a Part (SDK for Node.js)

Function

After a multipart upload is initiated, this API uploads a part to a specified bucket. In the upload request, the multipart upload ID must be included.

When uploading a part, you must specify its upload ID and part number. You can select any part number between **1** and **10000**. A part number uniquely identifies a part and its location in the object you are uploading. If the number of an uploaded part is used to upload a new part, the uploaded part will be overwritten. Whenever you upload a part, OBS returns the ETag header in the response. For each part upload, you must record the part number and the ETag value. These values are required in subsequent requests to complete a multipart upload.

Restrictions

- To assemble parts, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Except for the last part uploaded, the other parts range from 0 to 5 GB, and the last part ranges from 100 KB to 5 GB. Part sizes will not be verified during upload because whether an uploaded part is the last one cannot be determined. Their sizes will be verified when being assembled.
- Each part is numbered from 1 to 10000.
- The minimum part size supported by an OBS 3.0 bucket is 100 KB, and that supported by an OBS 2.0 bucket is 5 MB. You are advised to perform the multipart upload on OBS 3.0 buckets.

Method

```
ObsClient.uploadPart(params)
```

Request Parameters

Table 9-8 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
PartNumber	number	Yes	Explanation: Part number. Restrictions: None Value range: An integer ranging from 1 to 10000. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
UploadId	string	Yes	Explanation: Multipart upload ID, which is generated by initiating a multipart upload. Restrictions: None Value range: The value must contain 32 characters, for example, 000001648453845DBB78F2340DD460D8. Default value: None
ContentMD5	string	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Restrictions: Base64-encoded, 128-bit MD5 value of the request body. Value range: Base64-encoded, 128-bit MD5 value of the request body calculated based on the RFC 1864 standard. Example: n58IG6hfM7vqI4K0vnWpog== Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Body	string stream.Readable	No	Explanation: Data stream of the object to be uploaded. Restrictions: The content size in a single upload ranges from 0 to 5 GB. Value range: None Default value: None
SourceFile	string	No	Explanation: Source file path of the object to be uploaded Restrictions: • The content size in a single upload ranges from 0 to 5 GB. • Body and SourceFile cannot be used together. • If both Body and SourceFile are left blank, the size of the uploaded object is 0 bytes. • Offset, PartSize, and SourceFile can be used together to specify part of the source file to upload. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Offset	number	No	Explanation: Start offset of a part in the source file. Restrictions: Offset , PartSize , and SourceFile can be used together to specify part of the source file to upload. Value range: A non-negative integer smaller than the size of the object to be uploaded, in bytes Default value: 0
PartSize	number	No	Explanation: Part size. Restrictions: Offset , PartSize , and SourceFile can be used together to specify part of the source file to upload. Value range: The value ranges from 100 KB to 5 GB, in bytes. Default value: 102400
SseKms	string	Yes when SSE-KMS is used	Explanation: SSE-KMS is used for encrypting objects on the server side. Restrictions: Currently, only KMS is supported. Value range: kms Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseKmsKey	string	No	Explanation: ID of the KMS master key when SSE-KMS is used. Restrictions: Valid value formats are as follows: 1. <i>regionID:domainID:key/key_id</i> 2. <i>key_id</i> In the preceding formats: • <i>regionID</i> indicates the ID of the region where the key is used. You can obtain it from Regions and Endpoints. • <i>domainID</i> indicates the ID of the account that the key is for. To obtain it, see How Do I Get My Account ID and User ID? (SDK for Node.js). • <i>key_id</i> indicates the ID of the key created on Data Encryption Workshop (DEW). To obtain it, see Viewing a CMK. Value range: None Default value: • If this parameter is not specified, the default master key will be used. • If there is not a default master key, OBS will create one and use it.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

Responses

Table 9-9 Responses

Type	Description
Table 9-10 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-10 .

Table 9-10 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-11 .

Parameter	Type	Description
InterfaceResult	Table 9-12	Explanation: Results outputted for a successful call. For details, see Table 9-12 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-11 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-12 UploadPartOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and becomes B when the object is downloaded, this indicates the contents of the object were changed. The ETag reflects changes of an object, not of the metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.

Code Examples

This example uploads a part to bucket **examplebucket** in a specified multipart upload (uploadId) of object **example/objectname**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function uploadPart() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the ID of the multipart upload.
      UploadId: "0000000xxxxxxx",
      // Specify the part number (1 in this example) of the part to upload. The part number is an integer
      // ranging from 1 to 10000.
    };
  }
}
```

```
PartNumber: 1,
// Specify the source file path of the part to upload (localfile in this example).
SourceFile: "localfile",
// Specify the part size (5 MB in this example), in bytes.
PartSize: 5 * 1024 * 1024,
// Specify the offset of the part (0 in this example).
Offset: 0
};
// Upload the part.
const result = await obsClient.uploadPart(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Upload part(%d) successful with bucket(%s) and object(%s)!", params.PartNumber,
params.Bucket, params.Key);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  console.log("ETag: %s", result.InterfaceResult.ETag);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
uploadPart();
```

9.4 Assembling Parts (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API assembles the uploaded parts to complete the multipart upload. Before performing this operation, you cannot download the uploaded data. When assembling parts, you need to copy the additional message header information recorded during the multipart upload initiation to the object metadata. Such information is processed in the same way the information in a regular object upload is processed. In the case of assembling parts concurrently, the Last Write Wins strategy applies, but the time of Last Write is defined as the time when a multipart upload was initiated.

As long as the multipart upload is not aborted, all uploaded parts occupy the space. However, after you assembled the specified parts, those uploaded but not assembled will be deleted to free up space.

When assembling parts, OBS arranges parts in ascending order by part number. If any object metadata is provided during the initiation of the multipart upload, OBS will associate this metadata with the object. When the multipart upload is complete, the individual parts will no longer exist. A part assembling request must

contain the upload ID, part numbers, and ETag values. OBS responses include the ETag that uniquely identifies the object data. This ETag is not required to be the MD5 hash value of the object data.

Restrictions

- To assemble parts, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- After a multipart upload is complete, the uploaded parts that were not assembled will be automatically deleted and cannot be restored. Before assembling parts, use the API for listing uploaded parts to check all parts to ensure no parts were left out.
- If the size of any part other than the last part is smaller than 100 KB, OBS returns **400 Bad Request**.

Method

```
ObsClient.completeMultipartUpload(params)
```

Request Parameters

Table 9-13 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
UploadId	string	Yes	Explanation: Multipart upload ID, which is generated by initiating a multipart upload . Restrictions: None Value range: The value must contain 32 characters, for example, 000001648453845DBB78F2340DD460D8 . Default value: None
Parts	Part[]	Yes	Explanation: List of parts to be assembled. For details, see Table 9-14 .

Parameter	Type	Mandatory (Yes/No)	Description
EncodingType	string	No	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Table 9-14 Part

Parameter	Type	Description
PartNumber	number	Explanation: Part number. Restrictions: None Value range: An integer ranging from 1 to 10000. Default value: None
ETag	string	Explanation: ETag of a part. It is calculated by encoding the 128-bit MD5 value of the part using Base64. Restrictions: ETag (MD5 value of the part data) can be obtained from the response returned by the API for uploading a part . Value range: The value must contain 32 characters. Default value: None

Responses

Table 9-15 Responses

Type	Description
Table 9-16 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-16 .

Table 9-16 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-17 .
InterfaceResult	Table 9-18	Explanation: Results outputted for a successful call. For details, see Table 9-18 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-17 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-18 CompleteMultipartUploadOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an assembled object calculated based on the ETag of each part. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and becomes B when the object is downloaded, this indicates the contents of the object were changed. The ETag reflects changes of an object, not of the metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
Bucket	string	Explanation: Name of the bucket in which parts are assembled
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .

Parameter	Type	Description
Location	string	Explanation: URL of the object obtained from assembling the parts. Example: https://example-Bucket.obs.regions.myhuaweicloud.com/example-Object
VersionId	string	Explanation: Version ID of the object obtained from assembling the parts. If versioning is enabled for the bucket, the object version ID will be returned.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==
EncodingType	string	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Code Examples

This example assembles the uploaded parts to complete multipart upload **00000188677110424014075CC4A77xxx** of object **example/objectname**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function completeMultipartUpload() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the multipart upload ID (00000188677110424014075CC4A77xxx in this example).
      UploadId: "00000188677110424014075CC4A77xxx",
    };
    // Specify the list of parts to be assembled. The parts must be sorted by part number in ascending order,
    // and the parts can be inconsecutive.
    Parts: [
      { PartNumber: 1, ETag: "etag1" },
      { PartNumber: 2, ETag: "etag2" },
      { PartNumber: 3, ETag: "etag3" },
    ]
  };
  // Assemble parts.
  const result = await obsClient.completeMultipartUpload(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Complete multipart upload successful with bucket(%s) and object(%s)!", params.Bucket,
      params.Key);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    return;
  }
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
  response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
  attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
}
};

completeMultipartUpload();
```

9.5 Listing Uploaded Parts (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API lists the uploaded parts in a specified bucket. This request must contain the multipart upload ID.

You can list the uploaded parts of a specified multipart upload or of all ongoing multipart uploads. For each listing request, OBS can list a maximum of 1,000 uploaded parts. If more than 1,000 parts were uploaded for a multipart upload, you need to send more than one request to list all uploaded parts. Assembled parts will not be listed.

Restrictions

- To list uploaded parts, you must be the bucket owner or have the required permission (**obs:object:ListMultipartUploadParts** in IAM or **ListMultipartUploadParts** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- A maximum of 1,000 parts can be listed each time. If the upload of a specified ID contains more than 1,000 parts, **InterfaceResult.IsTruncated** in the response is **true**, indicating not all parts were listed. In such case, you can use **InterfaceResult.NextPartNumberMarker** to obtain the start position for the next listing.

Method

```
ObsClient.listParts(params)
```

Request Parameters

Table 9-19 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
UploadId	string	Yes	Explanation: Multipart upload ID, which is generated by initiating a multipart upload. Restrictions: None Value range: The value must contain 32 characters, for example, 000001648453845DBB78F2340DD460D8. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
PartNumberMarker	number	No	Explanation: Part number the listing starts from. Restrictions: None Restrictions: OBS only lists parts with greater numbers than that specified by this parameter. Default value: None
MaxParts	number	No	Explanation: Maximum number of parts that can be listed per page. Restrictions: If the specified value is greater than 1000, only 1,000 parts are returned. Value range: The value ranges from 1 to 1000. Default value: 1000
EncodingType	string	No	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key. Restrictions: None Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Responses

Table 9-20 Responses

Type	Description
Table 9-21 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-21 .

Table 9-21 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-22 .
InterfaceResult	Table 9-23	Explanation: Results outputted for a successful call. For details, see Table 9-23 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-22 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-23 ListMultipartUploadsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Bucket	string	Explanation: Bucket name
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
UploadId	string	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 .
Initiator	Initiator	Explanation: Initiator of the multipart upload. For details, see Table 9-24 .
Owner	Owner	Explanation: Owner of the multipart upload, which is consistent with an initiator. For details, see Table 9-25 .
StorageClass	StorageClass Type	Explanation: Storage class of the object obtained from assembling parts. For details about storage classes, see Table 9-26 .

Parameter	Type	Description
PartNumberMarker	number	Explanation: Part number the listing begins from, which is consistent with that specified in the request.
NextPartNumberMarker	number	Explanation: Part number to start with for the next part listing request. This parameter is returned if some parts were not listed. You can set PartNumberMarker to this value in the next request to list the remaining parts.
MaxParts	number	Explanation: Maximum number of parts that can be listed per page, which is consistent with that specified in the request.
IsTruncated	boolean	Explanation: Whether all results are returned in the response. Value range: • true: Not all results are returned. • false: All results are returned.
Parts	Part []	Explanation: List of uploaded parts. For details, see Table 9-27 .
EncodingType	string	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key .

Table 9-24 Initiator

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the initiator Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the initiator Restrictions: The account name can contain 6 to 32 characters and must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed. Default value: None

Table 9-25 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 9-26 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Table 9-27 Part

Parameter	Type	Description
PartNumber	number	Explanation: Part number
LastModified	string	Explanation: Time when a part was last modified, in UTC
ETag	string	Explanation: ETag of a part. It is calculated by encoding the 128-bit MD5 value of the part using Base64.
Size	number	Explanation: Size of the space occupied by a part

Code Examples

This example lists the parts that have been uploaded for a multipart upload.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
```

```
coding may result in leakage.
// Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
// use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
// variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listParts() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the multipart upload ID (00000188677110424014075CC4A77xxx in this example).
      UploadId: "00000188677110424014075CC4A77xxx",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is
      // 1000. 100 is used in this example.
      MaxKeys: 100,
    };
    // List uploaded parts.
    const result = await obsClient.listParts(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List part successful with bucket(%s) and object(%s)!", params.Bucket, params.Key);
      console.log('RequestId: %s', result.CommonMsg.RequestId);
      console.log('Bucket: %s', result.InterfaceResult.Bucket);
      console.log('Key: %s', result.InterfaceResult.Key);
      console.log('UploadId: %s', result.InterfaceResult.UploadId);
      console.log('PartNumberMarker: %s', result.InterfaceResult.PartNumberMarker);
      console.log('NextPartNumberMarker: %s', result.InterfaceResult.NextPartNumberMarker);
      console.log('MaxParts: %s', result.InterfaceResult.MaxParts);
      console.log('IsTruncated: %s', result.InterfaceResult.IsTruncated);
      console.log('StorageClass: %s', result.InterfaceResult.StorageClass);
      console.log('Initiator[ID]: %s', result.InterfaceResult.Initiator['ID']);
      console.log('Owner[ID]: %s', result.InterfaceResult.Owner['ID']);
      for (let i = 0; i < result.InterfaceResult.Parts.length; i++) {
        const part = result.InterfaceResult.Parts[i];
        console.log("Part[%d]-ETag:%s, PartNumber:%d, LastModified:%s, Size:%d",
          i, part.PartNumber, part.LastModified, part.ETag, part.Size);
      }
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an
error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

listParts();
```

9.6 Listing Multipart Uploads (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API lists multipart uploads that have been initiated but not completed or cancelled.

You can list all the multipart uploads that are not assembled or aborted in a bucket. Each request can return up to 1,000 multipart uploads. If a bucket contains more than 1,000 multipart uploads, **InterfaceResult.IsTruncated** in the response is **true**, indicating not all uploads were listed. In such case, you can use **InterfaceResult.NextKeyMarker** and **InterfaceResult.NextUploadIdMarker** to obtain the start position for the next listing.

Restrictions

- To list multipart uploads, you must be the bucket owner or have the required permission (**obs:bucket:ListBucketMultipartUploads** in IAM or **ListBucketMultipartUploads** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.listMultipartUploads(params)
```


Request Parameters

Table 9-28 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Prefix	string	No	Explanation: Prefix that the object names in the multipart uploads to be listed must contain. Assume that you have the following objects: logs/day1, logs/day2, logs/day3, and ExampleObject.jpg. If you specify logs/ as the prefix, the multipart uploads that contain logs/day1, logs/day2, or logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all multipart uploads in the bucket will be returned. Restrictions: The value can contain 1 to 1,024 characters. Value range: None Default value: None
MaxUploads	number	No	Explanation: Maximum number of multipart uploads to list. Restrictions: If the specified value is greater than 1000, only 1,000 multipart uploads are returned. Value range: An integer from 1 to 1000 Default value: 1000

Parameter	Type	Mandatory (Yes/No)	Description
Delimiter	string	No	Explanation: Object names are grouped by this parameter, which is often used with Prefix. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes. Assume that a bucket has objects abcd, abcde, and bbcde in it. If Delimiter is set to d and Prefix is set to a, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcde as the prefix. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
KeyMarker	string	No	Explanation: Object name to start with when listing multipart uploads Restrictions: None Value range: The value of NextKeyMarker in the response body of the last request. Default value: None
UploadIdMarker	string	No	Explanation: Upload ID to start with when listing multipart uploads Restrictions: This parameter is valid only when used with KeyMarker. If both parameters are specified, multipart uploads with IDs greater than the specified UploadIdMarker for the specified KeyMarker are listed. Value range: The value of NextUploadIdMarker in the response body of the last request Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
EncodingType	string	No	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key . Restrictions: None Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Responses

Table 9-29 Responses

Type	Description
Table 9-30 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-30 .

Table 9-30 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-31 .

Parameter	Type	Description
InterfaceResult	Table 9-32	Explanation: Results outputted for a successful call. For details, see Table 9-32 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-31 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-32 ListMultipartUploadsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Parameter	Type	Description
Bucket	string	Explanation: Bucket name
KeyMarker	string	Explanation: Object name to start with when listing multipart uploads. This parameter corresponds to KeyMarker in the request.
UploadIdMarker	string	Explanation: Upload ID to start with when listing multipart uploads. This parameter corresponds to UploadIdMarker in the request.
NextKeyMarker	string	Explanation: Object name to start with for the next multipart upload listing request. This parameter is returned if some results were not returned. You can set KeyMarker in the next request to the returned value to list the remaining results.
NextUploadIdMarker	string	Explanation: Upload ID to start with for the next multipart upload listing request. This parameter is used together with NextKeyMarker . This parameter is returned if some listing results were not returned. You can set UploadIdMarker of the next request to the returned value to list the remaining multipart uploads.
MaxUploads	number	Explanation: Maximum number of multipart uploads that can be listed. This parameter corresponds to MaxUploads in the request.
IsTruncated	boolean	Explanation: Whether all results are returned in the response. Value range: • true: Not all results are returned. • false: All results are returned.

Parameter	Type	Description
Prefix	string	Explanation: Object name prefix in multipart uploads. This parameter corresponds to Prefix in the request. It defines the prefix that the object names in the multipart uploads to be listed must contain. Assume that you have the following objects: logs/day1 , logs/day2 , logs/day3 , and ExampleObject.jpg . If you specify logs/ as the prefix, the multipart uploads that contain logs/day1 , logs/day2 , or logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all multipart uploads in the bucket will be returned.
Delimiter	string	Explanation: Used to group object names involved in multipart uploads. This parameter corresponds to Delimiter in the request. This parameter is often used with Prefix . If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes . If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes . Assume that a bucket has objects abcd , abcde , and bbcde in it. If Delimiter is set to d and Prefix is set to a , objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d , objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcd as the prefix.
Uploads	Upload[]	Explanation: List of multipart uploads. For details, see Table 9-33 .

Parameter	Type	Description
EncodingType	string	Explanation: Encoding type for Key in the response. If Key in the response contains control characters that are not supported by the XML 1.0 standard, you can specify this parameter to encode Key .

Table 9-33 Upload

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
UploadId	string	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 .
Initiator	Initiator	Explanation: Initiator of the multipart upload. For details, see Table 9-34 .
Owner	Owner	Explanation: Account ID of the object owner. For details, see Table 9-35 .
StorageClass	StorageClassType	Explanation: Storage class of the object involved in the multipart upload. For details about the storage classes, see Table 9-36 .
Initiated	string	Explanation: Time when the multipart upload was initiated

Table 9-34 Initiator

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the initiator Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the initiator Restrictions: The account name can contain 6 to 32 characters and must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed. Default value: None

Table 9-35 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 9-36 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

This example lists all multipart uploads for bucket **examplebucket** that have been initiated but not completed or aborted.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listMultipartUploads() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the maximum number of multipart uploads that can be listed. 10 is used in this example.
      MaxUploads: 10
    };
  }
}
```

```
// List multipart uploads.
const result = await obsClient.listMultipartUploads(params);
if (result.CommonMsg.Status <= 300) {
  console.log("List multipart uploads successful with bucket(%s)!", params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  for (let i = 0; i < output.Uploads.length; i++) {
    const upload = output.Uploads[i];
    console.log("Upload[%d]-OwnerId:%s, UploadId:%s, Key:%s, Initiated:%s",
      i, upload.Owner.ID, upload.UploadId, upload.Key, upload.Initiated);
  };
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
listMultipartUploads();
```

9.7 Copying a Part (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API uploads a part for a specified multipart upload by copying data to a specified bucket.

After initiating a multipart upload task, you can add parts for this task by copying part or all of an existing object in OBS.

Assume that you copy a source object and save it as **part1**. If there is already an existing part called **part1**, the new **part1** will overwrite the existing **part1**. Then, only the new **part1** can be listed. The old **part1** is deleted. To prevent accidental deletion, ensure there is no existing object with the same name as the part involved when using this API. The source object remains unchanged during the copy process.

Restrictions

- To copy a part, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

- You cannot determine whether the operation is successful solely based on **status_code** returned in the HTTP header. For example, code **200** indicates that the server has received and started to process the request. The copy operation is considered successful only when the response body contains an ETag.

Method

ObsClient.copyPart(params)

Request Parameters

Table 9-37 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Destination bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
PartNumber	number	Yes	Explanation: Part number. Restrictions: None Value range: Value range: 1–10000 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
UploadId	string	Yes	Explanation: Multipart upload ID, which is generated by initiating a multipart upload . Restrictions: None Value range: The value must contain 32 characters, for example, 000001648453845DBB78F2340DD460D8 . Default value: None
CopySource	string	Yes	Explanation: Names of the source bucket and object. If the source object has multiple versions, the versionId parameter is used to specify the desired version. Example: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A Restrictions: Full-width characters and percent signs (%) must be URL-encoded. Value range: None Default value: None

Responses

Table 9-38 Responses

Type	Description
Table 9-39 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-39 .

Table 9-39 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-40 .
InterfaceResult	Table 9-41	Explanation: Results outputted for a successful call. For details, see Table 9-41 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-40 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-41 CopyPartOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and becomes B when the object is downloaded, this indicates the contents of the object were changed. The ETag reflects changes of an object, not of the metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
LastModified	string	Explanation: Time when a part was last modified, in UTC
PartNumber	number	Explanation: Part number.
SseKms	string	Explanation: SSE-KMS is used for encrypting objects on the server side.
SseKmsKey	string	Explanation: ID of the KMS master key when SSE-KMS is used.
SseC	string	Explanation: SSE-C is used for encrypting objects on the server side.

Parameter	Type	Description
SseCKeyMd5	string	Explanation: MD5 value of the key used for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key. Restrictions: Base64-encoded MD5 value of the key, for example, 4XvB3tbNTN+tIEVa0/fGaQ==

Code Examples

This example copies a part from object **example/sourceobjectname** in bucket **sourcebucketname** to object **example/objectname** in bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyPart() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the names of the source bucket (sourcebucketname in this example) and source object
      // (example/sourceobjectname in this example).
      CopySource: "sourcebucketname/example/sourceobjectname",
      // Specify the part number (1 in this example).
      PartNumber: 1,
      // Specify the multipart upload ID (00000188677110424014075CC4A77xxx in this example).
      UploadId: "00000188677110424014075CC4A77xxx",
    };
    // Copy the part.
    const result = await obsClient.copyPart(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Complete multipart upload successful with bucket(%s) and object(%s)!", params.Bucket,
        params.Key);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
  }
}
```

```
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
copyPart();
```

9.8 Aborting a Multipart Upload (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API aborts a multipart upload by specifying its ID.

After a multipart upload is aborted, its upload ID cannot be used to upload any part. And the space occupied by all uploaded parts will be freed up. If any parts are being uploaded when you abort the upload, they may or may not be uploaded successfully. To free up the space occupied by all uploaded parts, you must abort the multipart upload after all parts have been uploaded.

Restrictions

- To abort a multipart upload, you must be the bucket owner or have the required permission (**obs:object:AbortMultipartUpload** in IAM or **AbortMultipartUpload** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.abortMultipartUpload(params)
```

Request Parameters

Table 9-42 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
input	Table 9-43	Yes	Explanation: Input parameters in the request for aborting a multipart upload. For details, see Table 9-43 .

Table 9-43 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
UploadId	string	Yes	Explanation: Multipart upload ID, which is generated by initiating a multipart upload . Restrictions: None Value range: The value must contain 32 characters, for example, 000001648453845DBB78F2340DD460D8 . Default value: None

Responses

Table 9-44 Responses

Type	Description
Table 9-45 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 9-45 .

Table 9-45 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 9-46 .
InterfaceResult	Table 9-47	Explanation: Results outputted for a successful call. For details, see Table 9-47 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 9-46 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 9-47 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example aborts multipart upload **00000188677110424014075CC4A77xxx**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function abortMultipartUpload() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the multipart upload ID (00000188677110424014075CC4A77xxx in this example).
      UploadId: "00000188677110424014075CC4A77xxx",
    };
    // Abort the multipart upload.
    const result = await obsClient.abortMultipartUpload(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Abort multipart upload successful!");
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}

abortMultipartUpload();
```

Helpful Links

- [Aborting a Multipart Upload](#)
- [OBS Error Codes](#)

10 Temporarily Authorized Access (SDK for Node.js)

10.1 Using a Temporary URL for Authorized Access (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

OBS allows you to create a URL whose Query parameters carry authentication information by specifying the AK and SK, HTTP method, and request parameters. You can provide this URL for other users to grant them temporary access. When generating a URL, you need to specify the validity period of the URL to restrict the access duration of visitors.

To grant other users the temporary permission to perform operations (such as upload), you need to generate a URL for the corresponding request (such as PUT) and provide it for users.

Restrictions

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- If a CORS or signature mismatch error occurs, troubleshoot the issue as follows:
 - a. If CORS is not configured, you need to configure CORS rules on OBS Console. For details, see [Configuring CORS](#).
 - b. If the signatures do not match, check whether signature parameters are missing or invalid. For details, see [Authentication of Signature in a URL](#).

For example, during an object upload, the backend uses **Content-Type** to calculate the signature and generate a signed URL, but if **Content-Type** is not set or is set to an invalid value when the frontend uses that URL, a CORS error occurs. To resolve this issue, ensure **Content-Type** remains consistent at the frontend and backend.

Method

```
ObsClient.createSignedUrlSync(params)
```

Request Parameters

Table 10-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Method	HttpMethodType	Yes	Explanation: HTTP method. For details, see Table 10-2 .

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
SpecialParam	SpecialParam	No	Explanation: Sub-resource to be accessed Restrictions: None Value range: See Table 10-3 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	number	No	Explanation: Expiration time of the signed URL. Restrictions: None Value range: - The value of this parameter for temporary credentials ranges from 0 to 86400, in seconds. - The value of this parameter for permanent keys ranges from 0 to 630720000, in seconds. Default value: 300
Headers	object	No	Explanation: Headers in the request. Restrictions: None Value range: None Default value: None
QueryParams	object	No	Explanation: Query parameters in the request. Restrictions: None Value range: None Default value: None

Table 10-2 HttpMethodType

Constant	Description
GET	HTTP GET request
POST	HTTP POST request

Constant	Description
PUT	HTTP PUT request
DELETE	HTTP DELETE request
HEAD	HTTP HEAD request
OPTIONS	HTTP OPTIONS request

Table 10-3 SpecialParam

Constant	Applicable API
storagePolicy	Specify or obtain bucket storage classes.
quota	Specify or obtain bucket quotas.
storageinfo	Obtain bucket storage information.
location	Obtain bucket locations.
acl	Specify or obtain bucket ACLs or object ACLs.
policy	Specify, obtain, or delete bucket policies.
cors	Specify, obtain, or delete bucket CORS configurations.
versioning	Specify or obtain bucket versioning statuses.
website	Specify, obtain, or delete bucket website configurations.
logging	Specify or obtain bucket logging settings.
lifecycle	Specify, obtain, or delete lifecycle rules of buckets.
notification	Specify or obtain the notification configurations of buckets.
tagging	Specify, obtain, or delete bucket tags.
append	Append data to an object.
delete	Batch delete objects.
versions	List object versions in buckets.
uploads	List or initiate multipart uploads in buckets.
restore	Restore Archive objects.

Responses

Table 10-4 Responses

Parameter	Type	Description
SignedUrl	string	Explanation: The signed URL that carries the authentication information.
ActualSignedRequestHeaders	object	Explanation: Headers that need to be included in the request initiated using the signed URL.

To access OBS using a temporary URL generated by the OBS Node.js SDK, perform the following steps:

Step 1 Call **ObsClient.createSignedUrlSync** to generate a signed URL.

Step 2 Use any HTTP library to make an HTTP/HTTPS request to OBS.

----End

CAUTION

If a CORS or signature mismatch error occurs, refer to the following steps to troubleshoot the issue:

1. If CORS is not configured, you need to configure CORS rules on OBS Console. For details, see [Configuring CORS](#).
2. If the signatures do not match, check whether signature parameters are correct by referring to [Authentication of Signature in a URL](#). For example, during an object upload, the backend uses **Content-Type** to calculate the signature and generate an authorized URL, but if **Content-Type** is not set or set incorrectly when the frontend uses the authorized URL, a CORS error occurs. To avoid this issue, ensure that **Content-Type** fields at both the frontend and backend are kept consistent.

The following code examples use a temporary URL to create a bucket or upload, download, list, or delete objects.

Code Examples: Creating a Bucket

This example creates a bucket using a temporary URL.

```
// Import the OBS library.
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
var https = require('https');
var urlLib = require('url');
var crypto = require('crypto');
```

```
// Create an ObsClient instance.
var obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using
  // hard coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  server : 'https://your-endpoint'
});

let bucketName = 'bucketname';
let method = 'PUT';
let res = obsClient.createSignedUrlSync({Method : method, Bucket : bucketName});
let location = 'your-location';
let content = `<CreateBucketConfiguration><LocationConstraint>${location}</LocationConstraint></CreateBucketConfiguration>`;

// Make a PUT request to create a bucket.
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method : method,
  host : url.hostname,
  port : url.port,
  path : url.path,
  rejectUnauthorized : false,
  headers : res.ActualSignedRequestHeaders || {}
});

console.log('Creating bucket using url:' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {

    if(serverback.statusCode < 300){
      console.log('Creating bucket using temporary signature succeed.');
```

Code Examples: Uploading an Object

This example uploads an object using a temporary URL.

```
// Import the OBS library.
// Use npm to install the client.
```

```
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

function createSignedUrlSync() {
  // Specify the HTTP method. PUT is used in this example.
  const method = 'PUT';
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket",
    // Specify an object (example/objectname in this example).
    Key: "example/objectname",
    // Specify the HTTP method.
    Method: method,
    // Specify the validity period of the signed URL, in seconds (3600 in this example).
    Expires: 3600,
    // Specify the headers carried in the request.
    Headers: {
      "Content-Type": "text/plain",
    }
  };
};

// Create a signed URL for uploading an object.
const res = obsClient.createSignedUrlSync(params);
console.log("SignedUrl: %s", res.SignedUrl);
console.log("ActualSignedRequestHeaders: %v", res.ActualSignedRequestHeaders);

let content = 'Hello OBS';
// Make a PUT request to upload an object.
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
console.log('Creating object using url.' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {
    if (serverback.statusCode < 300) {
      console.log('Creating object using temporary signature succeed.');
```

```
    buffers = Buffer.concat(buffers);
    if (buffers.length > 0) {
        console.log(buffers.toString());
    };
    console.log("\n");
});
}).on('error', (err) => {
    console.log('Creating object using temporary signature failed!');
    console.log(err);
    console.log("\n");
});
if (content) {
    req.write(content);
};
req.end();
};

createSignedUrlSync();
```

Code Examples: Downloading an Object

This example downloads an object using a temporary URL.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
    // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
    // coding may result in leakage.
    // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
    // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
    // variables or import an AK/SK pair in other ways.
    // security_token: process.env.SECURITY_TOKEN,
    // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
    // in this example. Replace it with the one currently in use.
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

function createSignedUrlSync() {
    // Specify the HTTP method. GET is used in this example.
    const method = 'GET';

    const params = {
        // Specify the bucket name.
        Bucket: "examplebucket",
        // Specify an object (example/objectname in this example).
        Key: "example/objectname",
        // Specify the HTTP method.
        Method: method,
        // Specify the validity period of the signed URL, in seconds (3600 in this example).
        Expires: 3600,
    };

    // Make a GET request to download an object.
    var url = urlLib.parse(res.SignedUrl);
    var req = https.request({
        method: method,
        host: url.hostname,
        port: url.port,
        path: url.path,
        rejectUnauthorized: false,
```

```
headers: res.ActualSignedRequestHeaders || {}  
});  
console.log('Creating object using url:' + res.SignedUrl);  
  
req.on('response', (serverback) => {  
  var buffers = [];  
  serverback.on('data', (data) => {  
    buffers.push(data);  
  }).on('end', () => {  
    if (serverback.statusCode < 300) {  
      console.log('Getting object using temporary signature succeed.');    } else {  
      console.log('Getting object using temporary signature failed!');  
      console.log('status:' + serverback.statusCode);  
      console.log("\n");  
    };  
    buffers = Buffer.concat(buffers);  
    if (buffers.length > 0) {  
      console.log(buffers.toString());  
    };  
    console.log("\n");  
  });  
}).on('error', (err) => {  
  console.log('Getting object using temporary signature failed!');  
  console.log(err);  
  console.log("\n");  
});  
req.end();  
};  
  
createSignedUrlSync();
```

Code Examples: Listing Objects

This example lists objects using a temporary URL.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
  // coding may result in leakage.  
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
  access_key_id: process.env.ACCESS_KEY_ID,  
  secret_access_key: process.env.SECRET_ACCESS_KEY,  
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
  // variables or import an AK/SK pair in other ways.  
  // security_token: process.env.SECURITY_TOKEN,  
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here  
  // in this example. Replace it with the one currently in use.  
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
function createSignedUrlSync() {  
  // Specify the HTTP method. GET is used in this example.  
  const method = 'GET';  
  
  const params = {  
    // Specify the bucket name.  
    Bucket: "examplebucket",  
    // Specify the HTTP method.  
    Method: method,  
    // Specify the validity period of the signed URL, in seconds (3600 in this example).  
  };  
}
```

```
Expires: 3600,
};

// Make a GET request to obtain the object list.
var url = urlLib.parse(res.SignedUrl);
var req = https.request({
  method: method,
  host: url.hostname,
  port: url.port,
  path: url.path,
  rejectUnauthorized: false,
  headers: res.ActualSignedRequestHeaders || {}
});
console.log('Listing object using url:' + res.SignedUrl);

req.on('response', (serverback) => {
  var buffers = [];
  serverback.on('data', (data) => {
    buffers.push(data);
  }).on('end', () => {
    if (serverback.statusCode < 300) {
      console.log('Listing object using temporary signature succeed.');
```

Code Examples: Deleting an Object

This example deletes an object using a temporary URL.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});
```

```
function createSignedUrlSync() {
  // Specify the HTTP method. DELETE is used in this example.
  const method = 'DELETE';
  const params = {
    // Specify the bucket name.
    Bucket: "examplebucket",
    // Specify an object (example/objectname in this example).
    Key: "example/objectname",
    // Specify the HTTP method.
    Method: method,
    // Specify the validity period of the signed URL, in seconds (3600 in this example).
    Expires: 3600,
  };

  // Make a DELETE request to delete the object.
  var url = urlLib.parse(res.SignedUrl);
  var req = https.request({
    method: method,
    host: url.hostname,
    port: url.port,
    path: url.path,
    rejectUnauthorized: false,
    headers: res.ActualSignedRequestHeaders || {}
  });
  console.log('Deleting object using url.' + res.SignedUrl);
  req.on('response', (serverback) => {
    var buffers = [];
    serverback.on('data', (data) => {
      buffers.push(data);
    }).on('end', () => {
      if (serverback.statusCode < 300) {
        console.log('Deleting object using temporary signature succeed!');
      } else {
        console.log('Deleting object using temporary signature failed!');
        console.log('status:' + serverback.statusCode);
        console.log('\n');
      };
      buffers = Buffer.concat(buffers);
      if (buffers.length > 0) {
        console.log(buffers.toString());
      };
      console.log('\n');
    });
  }).on('error', (err) => {
    console.log('Deleting object using temporary signature failed!');
    console.log(err);
    console.log('\n');
  });
  req.end();
};

createSignedUrlSync();
```

11 Versioning (SDK for Node.js)

11.1 Versioning Overview (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

You can use versioning to store multiple versions of an object in a bucket.

When versioning is enabled for a bucket, OBS keeps multiple versions of an object in the bucket, allowing you to easily retrieve and restore historical versions or recover data in the event of accidental changes or application failures.

By default, versioning is disabled for new OBS buckets. In this case, if a newly uploaded object is using the name of the previously uploaded one, the new object will overwrite the previous one.

For details, see [Versioning](#).

11.2 Configuring Versioning for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can enable versioning to automatically maintain previous versions of an object. When versioning is enabled, you can access earlier versions of an object to

recover your data in the event of accidental actions or application failures. For more information, see [Versioning](#).

This API configures versioning for a bucket.

Restrictions

- To configure versioning for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketVersioning** in IAM or **PutBucketVersioning** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketVersioning(params)
```

Request Parameters

Table 11-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
VersionStatus	string	Yes	Explanation: Versioning status of the bucket. Restrictions: None Value range: Versioning status of the bucket. Possible values are: • Enabled • Suspended Default value: None

Responses

Table 11-2 Responses

Type	Description
Table 11-3 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-3 .

Table 11-3 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-4 .
InterfaceResult	Table 11-5	Explanation: Results outputted for a successful call. For details, see Table 11-5 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-4 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-5 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example configures versioning for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/
```

```
intl/en-us/usermanual-ca/ca_01_0003.html.  
access_key_id: process.env.ACCESS_KEY_ID,  
secret_access_key: process.env.SECRET_ACCESS_KEY,  
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
variables or import an AK/SK pair in other ways.  
// security_token: process.env.SECURITY_TOKEN,  
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here  
in this example. Replace it with the one currently in use.  
server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function setBucketVersioning() {  
  try {  
    const params = {  
      // Specify the bucket name.  
      Bucket: "examplebucket",  
      // Specify a versioning status (Enabled in this example) for the bucket.  
      VersionStatus: 'Enabled'  
    };  
  
    // Configure versioning for the bucket.  
    const result = await obsClient.setBucketVersioning(params);  
    if (result.CommonMsg.Status <= 300) {  
      console.log("Set bucket(%s)'s versioning status successful!", params.Bucket);  
      console.log("RequestId: %s", result.CommonMsg.RequestId);  
      return;  
    }  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  }  
};  
  
setBucketVersioning();
```

11.3 Viewing the Versioning Status of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can enable versioning to automatically maintain previous versions of an object. When versioning is enabled, you can access earlier versions of an object to recover your data in the event of accidental actions or application failures. For more information, see [Versioning](#).

This API obtains the versioning status of a bucket.

Restrictions

- To view the versioning status of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketVersioning** in IAM or **GetBucketVersioning** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketVersioning(params)
```

Request Parameters

Table 11-6 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 11-7 Responses

Type	Description
Table 11-8 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-8 .

Table 11-8 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-9 .
InterfaceResult	Table 11-10	Explanation: Results outputted for a successful call. For details, see Table 11-10 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-9 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-10 GetBucketVersioningOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
VersionStatus	string	Explanation: Versioning status of the bucket

Code Examples

You can call **ObsClient.getBucketVersioning** to view the versioning status of a bucket. The following code shows how to view the versioning status of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketVersioning() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Obtain the versioning status of the bucket.
    const result = await obsClient.getBucketVersioning(params);
    if (result.CommonMsg.Status <= 300) {
```

```
console.log("Get bucket(%s)'s versioning status successful!", params.Bucket);
console.log("RequestId: %s", result.CommonMsg.RequestId);
console.log("VersionStatus: %s", result.InterfaceResult.VersionStatus);
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
getBucketVersioning();
```

11.4 Obtaining an Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API downloads an object version from OBS to your local computer.

Restrictions

- To download an object, you must be the bucket owner or have the required permission (**obs:object:GetObject** in IAM or **GetObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- Objects in the Archive storage class can be downloaded only when they are in the **Restored** status.

Method

```
ObsClient.getObject(params)
```

Request Parameters

Table 11-11 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Restrictions: None Value range: The value must contain 32 characters. Default value: None
SaveAsFile	string	No	Explanation: The download path with the file name contained Restrictions: A file name must be included in the path, for example, aa/bb.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SaveAsStream	boolean	No	Explanation: Whether the object is returned as a readable stream Value range: • true: The object is returned as a readable stream. • false: The object is not returned as a readable stream. Default value: false
ProgressCallback	function	No	Explanation: Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second). Restrictions: None Value range: None Default value: None
IfMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is the same as the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfModifiedSince	string	No	Explanation: If the object was modified after the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
IfNoneMatch	string	No	Explanation: Preset ETag. If the ETag of the object to be downloaded is different from the preset ETag, the object is returned. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
IfUnmodifiedSince	string	No	Explanation: If the object was modified before the time specified by this parameter, its content is returned. Otherwise, an error code is returned. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None
Range	string	No	Explanation: Download range. For example, 0-999 indicates the range is from byte 1 to byte 1000. Restrictions: The maximum length of Range is the length of the object minus 1. If the specified value exceeds this limit, the length of the object minus 1 is used. Value range: 0 to the object length minus 1. Format: bytes=x-y Default value: None
Origin	string	No	Explanation: Origin of the cross-domain request specified in the preflight request. It is usually a domain name. Restrictions: Each origin can contain at most one wildcard character (*). Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RequestHeader	string	No	Explanation: HTTP headers that can be used in cross-origin requests. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
ResponseCacheControl	string	No	Explanation: This parameter is used to rewrite the Cache-Control header in the response. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ResponseContentDisposition	string	No	Explanation: This parameter is used to rewrite the Content-Disposition header in the response. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseContentEncoding	string	No	Explanation: This parameter is used to rewrite the Content-Encoding header in the response. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ResponseContentLanguage	string	No	Explanation: This parameter is used to rewrite the Content-Language header in the response. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ResponseContent-Type	string	No	Explanation: This parameter is used to rewrite the Content-Type header in the response. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ResponseExpires	string	No	Explanation: This parameter is used to rewrite the Expires header in the response. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
ImageProcess	string	No	Explanation: Image processing command or style. Example for resizing and rotating an image: image/resize,m_fixed,w_100,h_100/rotate,90 Restrictions: None Value range: Command format: image/commands Style format: style/style-name For details about image processing parameters, see Processing Images . Default value: If no commands are entered, the original image is returned.
SseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for decrypting objects. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SseCKey	string	Yes when SSE-C is used	Explanation: Key used for decrypting objects when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None

 NOTE

- If a download request includes **IfUnmodifiedSince** or **IfMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **412 Precondition Failed**.
- If a download request includes **IfModifiedSince** or **IfNoneMatch** but does not meet the conditions specified by these parameters, an exception is thrown with HTTP status code **304 Not Modified**.

Responses

Table 11-12 Responses

Type	Description
Table 11-13 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-13 .

Table 11-13 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-14 .

Parameter	Type	Description
InterfaceResult	Table 11-15	Explanation: Results outputted for a successful call. For details, see Table 11-15 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-14 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-15 GetObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server.

Parameter	Type	Description
DeleteMarker	string	Explanation: Whether the deleted object is a delete marker.
LastModified	string	Explanation: Time when the object was last modified.
ContentLength	string	Explanation: Object size.
CacheControl	string	Explanation: Cache-Control header in the response.
ContentDisposition	string	Explanation: Content-Disposition header in the response.
ContentEncoding	string	Explanation: Content-Encoding header in the response.
ContentLanguage	string	Explanation: Content-Language header in the response.
ContentType	string	Explanation: MIME type of the object.
Expires	string	Explanation: Expires header in the response.
ETag	string	Explanation: ETag of the object.
VersionId	string	Explanation: Object version ID.
WebsiteRedirectLocation	string	Explanation: Location where an object is redirected. It is available when website hosting is configured for a bucket.
StorageClass	string	Explanation: Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	string	Explanation: Restoration status of the object in the Archive storage class.

Parameter	Type	Description
Expiration	string	Explanation: Expiration details.
Content	string stream.Re adable	Explanation: Content of the object. The content is left blank if SaveAsFile was specified. The content is a stream.Readable object if SaveAsStream was set to true . The content is a Buffer object if neither SaveAsFile nor SaveAsStream was specified.
Metadata	object	Explanation: Custom metadata of the object. You need to add in the CORS configurations the additional headers allowed to be carried in the response. For example, you can add the x-amz-meta-property1 header to allow custom metadata property1 to be returned.

Code Examples

You can call **ObsClient.getObject** to obtain a version of an object by specifying **VersionId**. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object to download (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the version ID of the object. If the version ID is not specified, the object of the latest version is
```

```
downloaded by default.
  VersionId: 'G001117FCE89978B0000401205D5DC9A'
};

// Download a version of an object.
const result = await obsClient.getObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

getObject();
```

11.5 Copying an Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API copies an object version in a specified bucket. You can copy an object of up to 5 GB in a single operation.

Restrictions

- To copy an object, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- This API requires carrying the information about the source bucket and object using headers. A message body cannot be carried.
- The target object size ranges from 0 to 5 GB. If the source object size exceeds 5 GB, you must use a multipart copying API by referring to [Copying a Part \(SDK for Node.js\)](#).

Method

```
ObsClient.copyObject(params)
```

Request Parameters

Table 11-16 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters.

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
CopySource	string	Yes	Explanation: Names of the source bucket and object. If the source object has multiple versions, the versionId parameter is used to specify the desired version. Example: source_bucket/sourceObject?versionId=G001117FCE89978B0000401205D5DC9A Restrictions: None Restrictions: Full-width characters and percent signs (%) must be URL-encoded. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ACL	AclType	No	Explanation: ACL that can be pre-defined during the object copy. For details about the ACL, see ACLs . Restrictions: None Value range: See Table 11-17 . Default value: None
MetadataDirective	MetadataDirectiveType	No	Explanation: Policy for copying the source object's attributes Restrictions: None Value range: See Table 11-18 . Default value: None
CopySourceIfMatch	string	No	Explanation: If the ETag of the source object is the same as the one specified by this parameter, it is copied. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceIfNone-Match	string	No	Explanation: If the ETag of the source object is different from the one specified by this parameter, it is copied. Otherwise, an error is returned. Restrictions: None Value range: The value must contain 32 characters. Default value: None
CopySourceIfUnmodifiedSince	string	No	Explanation: If the source object has not been modified since the specified time, it is copied. Otherwise, an exception is thrown. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceIfModifiedSince	string	No	Explanation: If the source object has been modified since the specified time, it is copied. Otherwise, an exception is thrown. Restrictions: The value must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
WebsiteRedirectLocation	string	No	Explanation: If the bucket is configured with website hosting, the request for obtaining the object can be redirected to another object in the bucket or an external URL. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL: WebsiteRedirectLocation:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata WebsiteRedirectLocation . Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection of objects that are in the root directory. Value range: None Default value: None
CopySourceSseC	string	Yes when SSE-C is used	Explanation: SSE-C is used for encrypting objects on the server side. Restrictions: Only AES256 is supported. Value range: AES256 Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CopySourceSseCKey	string	Yes when SSE-C is used	Explanation: Key used for encrypting the object when SSE-C is used Restrictions: A Base64-encoded, 256-bit value, for example, K7QkYpBkM5+hca27fsNkUnNVaobncnLht/rCB2o/9Cw= Value range: None Default value: None
CacheControl	string	No	Explanation: Cache-Control in the response is rewritten. Restrictions: None Value range: See the Cache-Control values defined in HTTP. Default value: None
ContentDisposition	string	No	Explanation: Content-Disposition in the response is rewritten. Restrictions: None Value range: See the Content-Disposition values defined in HTTP. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ContentEncoding	string	No	Explanation: Content-Encoding in the response is rewritten. Restrictions: None Value range: See the Content-Encoding values defined in HTTP. Default value: None
ContentLanguage	string	No	Explanation: Content-Language in the response is rewritten. Restrictions: None Value range: See the Content-Language values defined in HTTP. Default value: None
ContentType	string	No	Explanation: Content-Type in the response is rewritten. Restrictions: None Value range: See What Is Content-Type (MIME)? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Expires	string	No	Explanation: Expires in the response is rewritten. Restrictions: None Value range: See the Expires values defined in HTTP. Default value: None
StorageClass	StorageClassType	No	Explanation: Storage class configured when copying the object. Restrictions: None Value range: See Table 11-19 . Default value: None. If this parameter is not specified, the storage class of the bucket is used as that of the object.

Parameter	Type	Mandatory (Yes/No)	Description
Metadata	object	No	Explanation: Custom metadata of the target object. You can add a header starting with x-obs-meta- in the request to define metadata. The custom metadata will be returned in the response when you retrieve the object or query the object metadata. Restrictions: • The custom metadata cannot exceed 8 KB in total. To measure the size, calculate the sum of bytes of all UTF-8 encoded keys and values. • The custom metadata keys are case insensitive, but are stored in lowercase in OBS. The key values are case sensitive. • Both custom metadata keys and their values must conform to US-ASCII standards. If non-ASCII or unrecognizable characters are required, they must be encoded and decoded in URL or Base64 on the client, because the server does not perform such operations. Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantRead	string	No	Explanation: ID (domain_id) of an account the READ permission is granted to. The account with the READ permission can read the object to copy and obtain its metadata. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantReadAcp	string	No	Explanation: ID (domain_id) of an account the READ_ACP permission is granted to. The account with the READ_ACP permission can read the ACL of the object to copy. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
GrantWriteAcp	string	No	Explanation: ID (domain_id) of an account the WRITE_ACP permission is granted to. The account with the WRITE_ACP permission can modify the ACL of the object to copy. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None
GrantFullControl	string	No	Explanation: ID (domain_id) of an account the FULL_CONTROL permission is granted to. The account with the FULL_CONTROL permission can read the object to copy, obtain its metadata, and obtain and modify its ACL. Restrictions: Use commas (,) to separate multiple accounts. Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
SuccessActionRedirect	string	No	Explanation: Address (URL) to which a successfully answered request is redirected. If the value is valid and the request is successful, OBS returns status code 303. Location contains the value of this parameter, as well as the bucket name, object name, and object ETag. If the value is invalid, OBS ignores this parameter. In such case, the Location header is the object address, and OBS returns a status code based on whether the operation succeeded or failed. Restrictions: None Value range: None Default value: None

Table 11-17 AclType

Constant	Default Value	Description
ObsClient.enums.AclPrivate	private	Private read and write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
ObsClient.enums.AclPublicRead	public-read	Public read and private write If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadWrite	public-read-write	Public read and write If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart upload tasks. If this permission is granted on an object, everyone can obtain the content and metadata of the object.
ObsClient.enums.AclPublicReadDelivered	public-read-delivered	Public read on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart tasks, metadata, and object versions and read the content and metadata of objects in the bucket. NOTE AclPublicReadDelivered does not apply to objects.

Constant	Default Value	Description
ObsClient.enums.AclPublicReadWriteDelivered	public-read-write-delivered	Public read and write on a bucket as well as the objects in the bucket. If this permission is granted on a bucket, anyone can read the object list, multipart uploads, metadata, and object versions in the bucket and can upload or delete objects, initiate multipart upload tasks, upload parts, assemble parts, copy parts, and abort multipart uploads. They can also read the content and metadata of objects in the bucket. NOTE AclPublicReadWriteDelivered does not apply to objects.
ObsClient.enums.AclBucketOwnerFullControl	bucket-owner-full-control	If this permission is granted on an object, only the bucket and object owners have the full control over the object. By default, if you upload an object to a bucket of any other user, the bucket owner does not have the permissions on your object. After you grant this policy to the bucket owner, the bucket owner can have full control over your object. For example, if user A uploads object x to user B's bucket, user B does not have the control over object x. If user A sets the bucket-owner-full-control policy for object x, user B then has the control over object x.

Table 11-18 MetadataDirectiveType

Constant	Default Value	Description
ObsClient.enums.CopyMetadata	COPY	When copying an object, the object's attributes are also copied. NOTICE This value is used only in the API for Copying an Object (SDK for Node.js) .
ObsClient.enums.ReplaceMetadata	REPLACE	REPLACE uses the complete header carried in the current request to replace the original one and deletes the metadata that is not specified.
ObsClient.enums.ReplaceNewMetadata	REPLACE_NEW	REPLACE_NEW replaces the metadata that already has a value, assigns a value to the metadata that does not have a value, and retains the metadata that is not specified. NOTICE This value is used only in the API for Configuring Object Metadata (SDK for Node.js) .

Table 11-19 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

 NOTE

- If **CopySourceIfUnmodifiedSince**, **CopySourceIfMatch**, **CopySourceIfModifiedSince**, or **CopySourceIfNoneMatch** is included but the specified condition is not met, **412 precondition failed** will be returned.
- **CopySourceIfModifiedSince** and **CopySourceIfNoneMatch** can be used together. So can **CopySourceIfUnmodifiedSince** and **CopySourceIfMatch**.

Responses

Table 11-20 Responses

Type	Description
Table 11-21 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-21 .

Table 11-21 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-22 .
InterfaceResult	Table 11-23	Explanation: Results outputted for a successful call. For details, see Table 11-23 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-22 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .

Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-23 CopyObjectOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server.
LastModified	string	Explanation: Time when the target object was last modified, in UTC.
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of the target object. The ETag is a unique identifier for the object's contents, used to determine if the object has been updated. For example, if the ETag value is A when an object is uploaded and becomes B when the object is downloaded, this indicates the contents of the object were changed. The ETag reflects changes of an object, not of the metadata. Objects created by the upload and copy operations have unique ETags after being encrypted using MD5. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.
CopySourceVersionId	string	Explanation: Version ID of the source object.

Parameter	Type	Description
VersionId	string	Explanation: Version ID of the target object.

Code Examples

You can call **ObsClient.copyObject** to copy a specific version of an object by specifying **versionId** in the **CopySource** parameter. Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an ObsClient instance.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function copyObject() {
  try {
    const params = {
      // Specify the target bucket name.
      Bucket: "examplebucket",
      // Specify the name of the object copy (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the source bucket name (sourcebucketname in this example), source object name
      // (sourceobjectkey in this example), and source object version ID (G001117FCE89978B0000401205D5DC9A
      // in this example).
      CopySource: 'sourcebucketname/sourceobjectname?versionId=G001117FCE89978B0000401205D5DC9A'
    };
    // Copy a version of an object.
    const result = await obsClient.copyObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Copy Object(bucket:%s, object: %s) successful from bucket/object: %s!", params.Bucket,
        params.Key, params.CopySource);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("ETag: %s, LastModified:%s", result.InterfaceResult.ETag,
        result.InterfaceResult.LastModified);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
      response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
      attempting to communicate with OBS, for example, the client was unable to access the network.");
  }
}
```

```
console.log(error);  
};  
};  
copyObject();
```

11.6 Restoring an Archive Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

To download an object in the Archive storage class, you need to restore it first. After an object is restored, a copy of the object is saved in the Standard storage class. By doing so, the object in the Archive storage class and its copy in the Standard storage class co-exist in the bucket. The copy will be automatically deleted once its retention period ends.

This API restores an Archive object in a specified bucket.

Restrictions

- To restore an Archive object, you must be the bucket owner or have the required permission (**obs:object:RestoreObject** in IAM or **RestoreObject** in a bucket policy.) For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Configuring an Object Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- To extend the validity period of the Archive data restored, you can repeatedly restore the data, but you will be billed for each restoration. After a restoration, the validity period of Standard object copies will be extended, and you need to pay for storing these copies during the extended period.

Method

```
ObsClient.restoreObject(params)
```

Request Parameters

Table 11-24 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt, the object name is folder/test.txt. Restrictions: The object specified in ObsClient.restoreObject must be in the Archive storage class. Otherwise, an error is reported. Value range: The value can contain 1 to 1,024 characters. Default value: None
Version Id	string	No	Explanation: Version ID of the Archive object to restore. Restrictions: None Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is specified.

Parameter	Type	Mandatory (Yes/No)	Description
Days	number	Yes	Explanation: After an object is restored, a Standard copy is generated for the object. This parameter specifies how long the Standard copy can be retained, that is, the validity period of the restored object. Restrictions: None Value range: The value ranges from 1 to 30, in days. Default value: None
Tier	RestoreTierType	No	Explanation: Tier of the restoration speed. You can select a suitable tier based on your needs. Restrictions: None Value range: See Table 11-25 . Default value: Standard

Table 11-25 RestoreTierType

Constant	Default Value	Description
ObsClient.enums.RestoreTierExpedited	Expedited	Objects can be quickly restored from Archive storage within 1 to 5 minutes.
ObsClient.enums.RestoreTierStandard	Standard	Objects can be restored from Archive storage within 3 to 5 hours.

Responses

Table 11-26 Responses

Type	Description
Table 11-27 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-27 .

Table 11-27 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-28 .
InterfaceResult	Table 11-29	Explanation: Results outputted for a successful call. For details, see Table 11-29 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-28 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-29 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example restores version **G001117FCE89978B0000401205D5DC9A** of Archive object **example/objectname** in bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function restoreObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the version ID of the object.
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
      // Specify how long the restored object will be retained, in days. The value ranges from 1 to 30 (1 is
      // used in this example).
      Days: 1,
      // Specify the restore speed (obs.RestoreTierExpedited in this example). By default, the object is
      // restored at an expedited speed.
    };
  }
}
```



```
Tier: obs.enums.RestoreTierExpedited
};
// Restore the Archive object.
const result = await obsClient.restoreObject(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Restore object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};

restoreObject();
```

**CAUTION**

To prolong the validity period of the Archive data restored, you can repeatedly restore the data, but you will be billed for each restoration. After a second restore, the validity period of Standard object copies will be prolonged, and you need to pay for storing these copies during the prolonged period.

11.7 Listing Object Versions in a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API lists some or all of the object versions in a bucket. When listing the object versions, you can specify the criteria such as the prefix, number, and start position. Returned object versions are listed in alphabetical order by object name.

Restrictions

- To list object versions in a bucket, you must be the bucket owner or have the required permission (**obs:bucket:ListBucketVersions** in IAM or **ListBucketVersions** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.listVersions(params)
```

Request Parameters

Table 11-30 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
KeyMarker	string	No	Explanation: Object name to start with when listing object versions in a bucket. All object versions following the specified value are listed in lexicographical order by object name. Restrictions: This parameter is only available for listing objects with multiple versions. Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionIdMarker	string	No	Explanation: Version ID you want to start listing from. All object versions following the value specified by this parameter are listed in alphabetical order by object name and version ID. This parameter must be used together with KeyMarker. KeyMarker specifies the object name, and VersionIdMarker specifies the version ID of the specified object. Restrictions: • This parameter is only available for listing objects with multiple versions. • If the object specified by KeyMarker does not have the version specified by VersionIdMarker, the value of this parameter is invalid. Value range: A version ID of the object specified by KeyMarker Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Prefix	string	No	Explanation: Prefix that the object names to be listed must contain. Assume that you have the following objects: logs/day1, logs/day2, logs/day3, and ExampleObject.jpg. If you specify logs/ as the prefix, logs/day1, logs/day2, and logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all objects in the bucket will be returned. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
MaxKeys	number	No	Explanation: The maximum number of objects returned in the response in alphabetical order Restrictions: None Value range: The value ranges from 1 to 1000. If the specified value is greater than 1000, only 1,000 objects are returned. Default value: 1000

Parameter	Type	Mandatory (Yes/No)	Description
Delimiter	string	No	Explanation: Object names are grouped by this parameter, which is often used with Prefix. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes. Assume that a bucket has objects abcd, abcde, and bbcde in it. If Delimiter is set to d and Prefix is set to a, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcd as the prefix. For a parallel file system, if this parameter is not specified, all the content in the directory is recursively listed by default, and subdirectories are also listed. In big data scenarios, parallel file systems usually have multiple directory levels, and each directory level has a large number of objects. In such case, you are advised to configure [Delimiter=/] to list only the content in the current directory, without the content in subdirectories, to make the listing more efficient. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
EncodingType	string	No	Explanation: Encoding type for some elements in the response. If Delimiter , KeyMarker , Prefix , CommonPrefixes , NextKeyMarker , or Key contain any control characters that are not supported by the XML 1.0 standard, you can configure EncodingType to encode these parameters in the response. Restrictions: None Value range: url Default value: None. If you leave this parameter blank, encoding is not applied.

Responses

Table 11-31 Responses

Type	Description
Table 11-32 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 11-32 .

Table 11-32 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 11-33 .

Parameter	Type	Description
InterfaceResult	ListVersionsOutput	Explanation: Results outputted for a successful call. For details, see Table 11-34 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 11-33 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 11-34 ListVersionsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server. Default value: None
Location	string	Explanation: Region where a bucket is located.
Name	string	Explanation: Bucket name.
Prefix	string	Explanation: Object name prefix, which is consistent with that set in the request. Assume that you have the following objects: logs/day1 , logs/day2 , logs/day3 , and ExampleObject.jpg . If you specify logs/ as the prefix, logs/day1 , logs/day2 , and logs/day3 will be returned. If this parameter is left blank and there are no other filtering criteria, all objects in the bucket will be returned.
KeyMarker	string	Explanation: Object name to start listing from. All object versions following the value specified by this parameter are listed in alphabetical order by object name. This parameter corresponds to KeyMarker in the request.
VersionIdMarker	string	Explanation: Version ID of the key marker object. Version ID you want to start listing from. All object versions following the value specified by this parameter are listed in alphabetical order by object name and version ID. This parameter must be used together with KeyMarker . KeyMarker specifies the object name, and VersionIdMarker specifies the version ID of the specified object.

Parameter	Type	Description
Delimiter	string	Explanation: Object names are grouped by this parameter, which is often used with Prefix. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefixes. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefixes. Assume that a bucket has objects abcd, abcde, and bbcde in it. If Delimiter is set to d and Prefix is set to a, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix. If only Delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefixes with abcd as the prefix, and bbcde is placed separately into another CommonPrefixes with bbcde as the prefix. For a parallel file system, if this parameter is not specified, all the content in the directory is recursively listed by default, and subdirectories are also listed. In big data scenarios, parallel file systems usually have multiple directory levels, and each directory level has a large number of objects. In such case, you are advised to configure [Delimiter=/] to list only the content in the current directory, without the content in subdirectories, to make the listing more efficient.
MaxKeys	number	Explanation: The maximum number of objects returned in the response in alphabetical order. This parameter corresponds to MaxKeys in the request.
IsTruncated	boolean	Explanation: Whether all results are returned in the response. A maximum of 1,000 objects can be listed at a time. If the number of objects is greater than 1,000, the objects beyond 1,000 cannot be returned. Value range: • true: Not all results are returned. • false: All results are returned.

Parameter	Type	Description
NextKeyMarker	string	Explanation: Object name to start with in the next request for listing object versions. This parameter is returned if some results were not returned. You can set KeyMarker in the next request to the returned value to list the remaining results.
NextVersionIdMarker	string	Explanation: Version ID to start with in the next request for listing object versions. It must be used together with NextKeyMarker . NextVersionIdMarker is returned when some results were not returned. You can set VersionIdMarker in the next request to the returned value to list the remaining results.
Versions	Version []	Explanation: List of object versions in the bucket. For details, see Table 11-36 .
DeleteMarkers	DeleteMarker []	Explanation: List of delete markers in the bucket. For details, see Table 11-37 .
CommonPrefixes	Table 11-35 []	Explanation: List of object name prefixes grouped based on the Delimiter parameter (if specified). For details, see CommonPrefix .
EncodingType	string	Explanation: Encoding type for some elements in the response. If Delimiter , KeyMarker , Prefix , CommonPrefixes , NextKeyMarker , or Key contain any control characters that are not supported by the XML 1.0 standard, you can configure EncodingType to encode these parameters in the response.

Table 11-35 CommonPrefix

Parameter	Type	Description
Prefix	string	Explanation: Prefix of objects in the bucket.

Table 11-36 Version

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing an object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt . Value range: The value can contain 1 to 1,024 characters. Default value: None
VersionId	string	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
LastModified	string	Explanation: Time when the object was last modified, in UTC. Value range: UTC time Default value: None

Parameter	Type	Description
ETag	string	Explanation: Base64-encoded, 128-bit MD5 value of an object. It uniquely identifies the content of an object and can be used to check the object integrity. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, it indicates that the object content has been changed. The ETag reflects changes to the contents of the object, not its metadata. An object created by an upload or copy operation has a unique ETag. Restrictions: If an object is encrypted using server-side encryption, the ETag is not the MD5 value of the object. Value range: The value must contain 32 characters. Default value: None
Size	number	Explanation: Object size, in bytes. Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
Owner	Owner	Explanation: Object owner. For details, see Table 11-38 .
StorageClass	StorageClassType	Explanation: Storage class of the object. For details, see Table 11-39 .
IsLatest	boolean	Explanation: Whether the object is the latest version. Value range: • true: The object is the latest version. • false: The object is not the latest version. Default value: false

Table 11-37 DeleteMarker

Parameter	Type	Description
Key	string	Explanation: Object name. An object is uniquely identified by an object name in a bucket. An object name is a complete path of the object that does not contain the bucket name. For example, if the address for accessing the object is examplebucket.obs.ap-southeast-1.myhuaweicloud.com/folder/test.txt , the object name is folder/test.txt .
VersionId	string	Explanation: Object version ID.
IsLatest	boolean	Explanation: Whether the object is the latest version. Value range: • true: The object is the latest version. • false: The object is not the latest version.
LastModified	string	Explanation: Time when the object was last modified, in UTC.
Owner	Owner	Explanation: Object owner. This parameter contains the domain ID and name of the object owner. For details, see Table 11-38 .
StorageClass	StorageClassType	Explanation: Storage class of the object. For details, see Table 11-39 .

Table 11-38 Owner

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	Yes if used as a request parameter	Explanation: Account (domain) ID of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and User ID? (SDK for Node.js) . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Display Name	string	No	Explanation: Account name of the owner Default value: None

Table 11-39 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples: Simple Listing

This example lists object versions in a bucket. A maximum of 1,000 object versions can be returned.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
```

```
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // List object versions in the bucket.
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // Obtain object versions.
      for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
          j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
      };
      // Obtain delete markers.
      for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
          j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listVersions();
```

NOTE

- A maximum of 1,000 object versions can be listed each time. If a bucket contains more than 1,000 object versions, **InterfaceResult.IsTruncated** in the response is **true**, indicating not all object versions were listed. In such case, you can use **InterfaceResult.NextKeyMarker** and **InterfaceResult.NextVersionIdMarker** to obtain the start position for the next listing.
- To obtain all object versions in a specified bucket at a time, refer to [Code Examples: Using Pagination to List All Object Versions](#).

Code Examples: Listing a Specified Number of Object Versions

This example lists a specified number of object versions in a bucket.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
```

```
// Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
coding may result in leakage.
// Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is
      // 1000. 100 is used in this example.
      MaxKeys: 100,
    };
    // List object versions in the bucket.
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // Obtain object versions.
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // Obtain delete markers.
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};

listVersions();
```

Code Examples: Listing Object Versions by Prefix

This example lists object versions with a specified prefix.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');
```



```
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will
      // be listed.
      Prefix: "test/",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is
      // 1000. 100 is used in this example.
      MaxKeys: 100,
    };

    // List object versions in the bucket.
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // Obtain object versions.
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // Obtain delete markers.
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[i];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
  };
};

console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

listVersions();
```

Code Examples: Specifying a Start Position for Listing

This example lists object versions from the specified position.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will
      // be listed.
      Prefix: "test/",
      // Specify the maximum number of objects to be returned in alphabetic order. The default value is
      // 1000. 100 is used in this example.
      MaxKeys: 100,
      // Specify the position (test/test2 in this example) where the object listing starts.
      KeyMarker: "test/test2",
    };
    // List object versions in the bucket.
    const result = await obsClient.listVersions(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("List version objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
        // Obtain object versions.
        for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
        };
        // Obtain delete markers.
        for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
          const val = result.InterfaceResult.Versions[j];
          console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
        };
      };
      return;
    };
  } catch (error) {
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  }
}
```

```
        console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
        console.log(error);  
    };  
};  
  
listVersions();
```

Code Examples: Using Pagination to List All Object Versions

This example lists all object versions using pagination.

```
// Import the OBS library.  
// Use npm to install the client.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code to install the client.  
// var ObsClient = require('./lib/obs');  
  
// Create an instance of ObsClient.  
const obsClient = new ObsClient({  
    // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard  
    coding may result in leakage.  
    // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.  
    access_key_id: process.env.ACCESS_KEY_ID,  
    secret_access_key: process.env.SECRET_ACCESS_KEY,  
    // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to  
    use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment  
    variables or import an AK/SK pair in other ways.  
    // security_token: process.env.SECURITY_TOKEN,  
    // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here  
    in this example. Replace it with the one currently in use.  
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"  
});  
  
async function listVersions() {  
    try {  
        const params = {  
            // Specify the bucket name.  
            Bucket: "examplebucket",  
            // Specify an encoding type. url is used in this example. If the objects to list contain special characters,  
            this parameter must be passed.  
            EncodingType: "url",  
        };  
        while (true) {  
            // List object versions in the bucket.  
            const result = await obsClient.listVersions(params);  
            if (result.CommonMsg.Status > 300) {  
                console.log("An ObsError was found, which means your request sent to OBS was rejected with an  
error response.");  
                console.log("Status: %d", result.CommonMsg.Status);  
                console.log("Code: %s", result.CommonMsg.Code);  
                console.log("Message: %s", result.CommonMsg.Message);  
                console.log("RequestId: %s", result.CommonMsg.RequestId);  
                return;  
            };  
            console.log("List version objects under the bucket(%s) successful!", params.Bucket);  
            console.log("RequestId: %s", result.CommonMsg.RequestId);  
            for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {  
                // Obtain object versions.  
                for (let i = 0; i < result.InterfaceResult.Versions.length; i++) {  
                    const val = result.InterfaceResult.Versions[i];  
                    console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',  
                        j, val.OwnerId, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);  
                };  
                // Obtain delete markers.  
                for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {  
                    const val = result.InterfaceResult.DeleteMarkers[i];  
                    console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
```

```
        j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
    };
    };
    if (result.InterfaceResult.IsTruncated === "true") {
        params.KeyMarker = result.InterfaceResult.NextKeyMarker;
        params.VersionIdMarker = result.InterfaceResult.NextVersionIdMarker;
    } else {
        break;
    };
    };
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
listVersions();
```

Code Examples: Listing All Object Versions in a Folder

All elements stored in OBS buckets are called objects. Folders are objects whose sizes are 0 and whose names end with a slash (/). You can set a folder name as a prefix to list objects in this folder. This example lists object versions in a folder.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
    // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
    // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
    // security_token: process.env.SECURITY_TOKEN,
    // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here in this example. Replace it with the one currently in use.
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function listVersions() {
    try {
        const params = {
            // Specify the bucket name.
            Bucket: "examplebucket",
            // Specify an object prefix (test/ in this example). Objects whose names contain the specified prefix will be listed.
            Prefix: "test/",
            // Specify an encoding type. url is used in this example. If the objects to list contain special characters, this parameter must be passed.
            EncodingType: "url",
        };
        while (true) {
            // List object versions in the bucket.
            const result = await obsClient.listVersions(params);
            if (result.CommonMsg.Status > 300) {
                console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
                console.log("Status: %d", result.CommonMsg.Status);
                console.log("Code: %s", result.CommonMsg.Code);
            }
        }
    } catch (error) {
        console.log(error);
    }
}
```

```
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
return;
};
console.log("List version objects under the bucket(%s) successful!", params.Bucket);
console.log("RequestId: %s", result.CommonMsg.RequestId);
for (let j = 0; j < result.InterfaceResult.Contents.length; j++) {
    // Obtain object versions.
    for (let j = 0; j < result.InterfaceResult.Versions.length; j++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, ETag:%s, Key:%s, VersionId, LastModified:%s, Size:%d',
            j, val.Owner.ID, val.ETag, val.Key, val.VersionId, val.LastModified, val.Size);
    };
    // Obtain delete markers.
    for (let i = 0; i < result.InterfaceResult.DeleteMarkers.length; i++) {
        const val = result.InterfaceResult.Versions[j];
        console.log('Content[%d]-OwnerId:%s, Key:%s, VersionId, LastModified:%s',
            j, val.Owner.ID, val.Key, val.VersionId, val.LastModified);
    };
};
if (result.InterfaceResult.IsTruncated === "true") {
    params.KeyMarker = result.InterfaceResult.NextKeyMarker;
    params.VersionIdMarker = result.InterfaceResult.NextVersionIdMarker;
} else {
    break;
};
};
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
listVersions();
```

11.8 Setting an ACL for an Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API calls **ObsClient.setObjectAcl** to set an ACL for an object version specified by the **VersionId** parameter. For details about the API definition, see [Configuring an Object ACL \(SDK for Node.js\)](#).

Code Examples

This example sets the ACL to allow all users to read object **example/objectname** from bucket **examplebucket** but only allow user **0a03f5833900d3730f13c00f49d5exxx** to write.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
```

```
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setObjectAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify the version ID of the object.
      VersionId: 'G001117FCE89978B0000401205D5DC9A',
      // Specify the owner of the object.
      Owner: { 'ID': 'ownerid' },
    };
    // Specify the information about the authorized user.
    Grants: [
      // Grant the write permission to a specified user (0a03f5833900d3730f13c00f49d5exxx in this
      // example).
      { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
      obsClient.enums.PermissionWrite },
      // Grant the read permission to all users.
      { Grantee: { Type: 'Group', URI: obsClient.enums.GroupAllUsers }, Permission:
      obsClient.enums.PermissionRead },
    ]
  };
  // Set the ACL.
  const result = await obsClient.setObjectAcl(params);
  if (result.CommonMsg.Status <= 300) {
    console.log("Set Object(%s)'s acl successful with Bucket(%s)!", params.Key, params.Bucket);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
  response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
  attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};

setObjectAcl();
```

NOTE

- For details about the error codes returned during the configuration of permissions for versioned objects, see [OBS Server-Side Error Codes \(SDK for Node.js\)](#).

11.9 Obtaining the ACL of an Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API calls **ObsClient.getObjectAcl** to obtain the ACL of an object version specified by the **VersionId** parameter. For details about the API definition, see [Obtaining the ACL of an Object \(SDK for Node.js\)](#).

Code Examples

This example obtains the ACL of version **G001117FCE89978B0000401205D5DC9A** of object **example/objectname** in bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObjectAcl() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
      // Specify the version ID of the object.
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
    };
    // Obtain the object ACL.
    const result = await obsClient.getObjectAcl(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s)'s acl successful with bucket(%s)!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Owner[ID]: %s', result.InterfaceResult.Owner.ID);
    }
  } catch (error) {
    console.error(error);
  }
}
```

```
console.log('Owner[Name]: %s', result.InterfaceResult.Owner.Name);
for (let i = 0; i < result.InterfaceResult.Grants.length; i++) {
  const grant = result.InterfaceResult.Grants[i];
  fmt.Printf("Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s\n",
    index, grant.Grantee.Type, grant.Grantee.ID, grant.Grantee.URI, grant.Permission);
};
return;
};
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};
getObjectAcl();
```

NOTE

- For details about the error codes returned during the configuration of permissions for versioned objects, see [OBS Server-Side Error Codes \(SDK for Node.js\)](#).

11.10 Deleting an Object Version (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can call **ObsClient.deleteObject** to delete an object version specified by the **VersionId** parameter. For details about the API definition, see [Deleting an Object \(SDK for Node.js\)](#).

You can also call **ObsClient.deleteObjects** to delete multiple object versions at a time by passing all of their **VersionId** values. For details about the API definition, see [Batch Deleting Objects \(SDK for Node.js\)](#).

Code Examples: Deleting an Object Version

This example deletes version **G001117FCE89978B0000401205D5DC9A** of object **example/objectname** from bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
```



```
coding may result in leakage.
// Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example) to delete.
      Key: 'example/objectname',
      // Specify the ID of the object version.
      VersionId: 'G001117FCE89978B0000401205D5DC9A'
    };
    // Delete a version of an object.
    const result = await obsClient.deleteObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

deleteObject();
```

Code Examples: Deleting Object Versions

This example deletes objects **objectname1**, **objectname2**, and **objectname3** from bucket **examplebucket** in a batch.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
```

```
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteObjects() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object list to delete.
      Objects: [
        { Key: 'objectname1', VersionId: "version1" },
        { Key: 'objectname2', VersionId: "version2" },
        { Key: 'objectname3', VersionId: "version3" }
      ]
    };
    // Delete the objects in a batch.
    const result = await obsClient.deleteObjects(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete objects under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      // Return details about which objects were deleted.
      console.log('Deleted:');
      for (let i = 0; i < result.InterfaceResult.Deleted.length; i++) {
        const deleted = result.InterfaceResult.Deleted[i];
        console.log("Deleted[%d]-Key:%s, VersionId:%s", i, deleted.Key, deleted.VersionId);
      };
      // Return details about which objects were not deleted.
      console.log('Errors:');
      for (let i = 0; i < result.InterfaceResult.Errors.length; i++) {
        const err = result.InterfaceResult.Errors[i];
        console.log("Errors[%d]-Key:%s, Code:%s", i, err.Key, err.Code);
      };
      console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
      console.log("Status: %d", result.CommonMsg.Status);
      console.log("Code: %s", result.CommonMsg.Code);
      console.log("Message: %s", result.CommonMsg.Message);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
      console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
      console.log(error);
    };
  };
}

deleteObjects();
```

12 Lifecycle (SDK for Node.js)

12.1 Configuring Lifecycle Rules for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes. For more information, see [Lifecycle Management](#).

This API configures lifecycle rules for a bucket.

NOTE

- Expired objects will be permanently deleted and cannot be restored.
- Multi-AZ redundancy is not available for Archive storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the Archive storage class based on a lifecycle rule.
- The minimum storage duration is 30 days for Infrequent Access storage and 90 days for Archive storage. After an object is transitioned to the Archive storage class, if it stays in this storage class for less than 90 days, you still need to pay for a full 90 days.

Restrictions

- There is no limit on the number of lifecycle rules in a bucket, but the total size of XML descriptions of all lifecycle rules in a bucket cannot exceed 20 KB.
- A maximum of 20 lifecycle rules can be configured for a parallel file system.
- To configure a lifecycle rule for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutLifecycleConfiguration** in IAM or **PutLifecycleConfiguration** in a bucket policy). For details, see

[Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketLifecycle(params)
```

Request Parameters

Table 12-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Rules	LifecycleRule[]	Yes	Explanation: Lifecycle rules of the bucket Restrictions: - There is no limit on the number of lifecycle rules in a bucket, but the total size of XML descriptions of all lifecycle rules in a bucket cannot exceed 20 KB. - A maximum of 20 lifecycle rules can be configured for a parallel file system. Value range: See Table 12-2 . Default value: None

Table 12-2 LifecycleRule

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	No if used as a request parameter	Explanation: Lifecycle rule ID. Restrictions: None Value range: The value must contain 1 to 255 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Prefix	string	Yes if used as a request parameter	Explanation: Object name prefix. It identifies the objects the rule applies to. You can leave this parameter blank to apply the rule to all objects in the bucket. Assume that you have the following objects: logs/day1, logs/day2, logs/day3, and ExampleObject.jpg. If you set Prefix to ExampleObject.jpg, the rule applies to object ExampleObject.jpg alone. If you set Prefix to logs/, the rule applies to objects logs/day1, logs/day2, and logs/day3. If you leave Prefix blank, the rule applies to all objects in the bucket. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
Status	string	Yes if used as a request parameter	Explanation: Whether the current rule is enabled. Restrictions: None Value range: Enabled Disabled Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Transitions	Transition []	No if used as a request parameter	Explanation: Policies for storage class transition, including transition time and the storage class after transition. Restrictions: This parameter is only available for the latest object version. Value range: See Table 12-3 . Default value: None
Expiration	Expiration	No if used as a request parameter	Explanation: Expiration time of an object. Restrictions: This parameter is only available for the latest object version. Value range: See Table 12-4 . Default value: None
NoncurrentVersionTransitions	NoncurrentVersion Transition []	No if used as a request parameter	Explanation: Policies for storage class transition of historical versions, including transition time and the storage class after transition. Restrictions: - This parameter is only available for historical object versions. - Versioning must be enabled (or suspended after being enabled) for the bucket. Value range: See Table 12-5 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
NoncurrentVersionExpiration	NoncurrentVersionExpiration	No if used as a request parameter	Explanation: Expiration time of historical object versions Restrictions: • This parameter is only available for historical object versions. • Versioning must be enabled (or suspended after being enabled) for the bucket. • This parameter is not available for parallel file systems. Value range: See Table 12-6 . Default value: None

 NOTE

Transitions, Expiration, NoncurrentVersionTransitions, and NoncurrentVersionExpiration cannot be left blank at the same time.

Table 12-3 Transition

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	Yes if used as a request parameter	Explanation: Storage class of the object after transition Restrictions: • The Standard storage class is not supported. • Restrictions on storage class transitions: – Only transitions from the Standard storage class to the Infrequent Access storage class are supported. To transition objects from Infrequent Access to Standard, you must manually do it. – Only transitions from the Standard or Infrequent Access storage class to the Archive storage class are supported. To transition objects from Archive to Standard or Infrequent Access, you must first restore these objects and then manually transition their storage classes. – Multi-AZ redundancy is not available for the Archive storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the Archive storage class based on a lifecycle rule. Value range: See Table 12-7. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Date	string	Yes if this parameter is used as a request parameter and Days is absent	Explanation: OBS executes the lifecycle rule for objects that were modified before the specified date. Restrictions: The value must conform with the ISO8601 standards and indicate UTC 00:00. For example, 2018-01-01T00:00:00.000Z indicates only objects that were last modified before the specified time are transitioned to the specified storage class. Value range: None Default value: None
Days	number	Yes if this parameter is used as a request parameter and Date is absent	Explanation: How many days can pass since the last update before the lifecycle rule takes effect Restrictions: This parameter is only available for the latest object version. Value range: 0 to $(2^{31} - 1)$, in days Default value: None

Table 12-4 Expiration

Parameter	Type	Mandatory (Yes/No)	Description
Date	string	Yes if this parameter is used as a request parameter and Days is absent	Explanation: OBS executes the lifecycle rule (deletion) for objects that were modified before the specified date. Restrictions: The value must conform with the ISO8601 standards and indicate UTC 00:00. For example, 2018-01-01T00:00:00.000Z indicates only objects that were last modified before the specified time are deleted. Value range: None Default value: None
Days	number	Yes if this parameter is used as a request parameter and Date is absent	Explanation: How many days can pass since the last update before the lifecycle rule (deletion) takes effect Restrictions: This parameter is only available for the latest object version. Value range: 1 to $(2^{31} - 1)$, in days Default value: None

Table 12-5 NoncurrentVersionTransition

Parameter	Type	Mandatory (Yes/No)	Description
StorageClass	StorageClassType	Yes if used as a request parameter	Explanation: Storage class of historical object versions after transition Restrictions: • The Standard storage class is not supported. • Restrictions on storage class transitions: – Only transitions from the Standard storage class to the Infrequent Access storage class are supported. To transition objects from Infrequent Access to Standard, you must manually do it. – Only transitions from the Standard or Infrequent Access storage class to the Archive storage class are supported. To transition objects from Archive to Standard or Infrequent Access, you must first restore these objects and then manually transition their storage classes. – Multi-AZ redundancy is not available for Archive storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the Archive storage class based on a lifecycle rule. Value range: See Table 12-7. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
NoncurrentDays	number	Yes if used as a request parameter	Explanation: Number of days an object is historical before the specified rule takes effect Restrictions: This parameter is only available for historical object versions. Value range: 0 to $(2^{31} - 1)$, in days Default value: None

Table 12-6 NoncurrentVersionExpiration

Parameter	Type	Mandatory (Yes/No)	Description
NoncurrentDays	number	Yes if used as a request parameter	Explanation: Number of days an object is historical before the specified rule takes effect Restrictions: This parameter is only available for historical object versions. Value range: 0 to $(2^{31} - 1)$, in days Default value: None

Table 12-7 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Responses

Table 12-8 Responses

Type	Description
Table 12-9 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 12-9 .

Table 12-9 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 12-10 .

Parameter	Type	Description
InterfaceResult	Table 12-11	Explanation: Results outputted for a successful call. For details, see Table 12-11 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 12-10 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 12-11 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples: Setting an Object Transition Policy

This example sets lifecycle rules to specify transition policies for latest and historical versions of objects in bucket **examplebucket**.

Code Examples: Setting an Object Expiration Time

This example sets lifecycle rules to specify expiration time for latest and historical versions of objects in bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLifecycle() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Lifecycle rules
      Rules: [
        {
          ID: 'rule1', Prefix: 'prefix1', Status: 'Enabled',
          // Specify that objects whose names contain the specified prefix will expire 60 days after creation.
          Expiration: { Days: 60 },
          // Specify that objects whose names contain the prefix will expire 60 days after becoming historical.
          NoncurrentVersionExpiration: { NoncurrentDays : 60 }
        },
        {
          ID: 'rule2', Prefix: 'prefix2', Status: 'Enabled',
          // Specify when the objects whose names contain the specified prefix will expire. The value must
          // conform to the ISO8601 standard. The time value must be 00:00:00 in UTC.
          Expiration: { Date: '2018-12-31T00:00:00Z' },
        }
      ]
    };
    // Configure lifecycles.
    const result = await obsClient.setBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s lifecycle configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
  }
}
```

```
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
setBucketLifecycle();
```

12.2 Obtaining the Lifecycle Rules of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes. For more information, see [Lifecycle Management](#).

This API returns the lifecycle rules of a bucket.

Restrictions

- To obtain the lifecycle rules of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetLifecycleConfiguration** in IAM or **GetLifecycleConfiguration** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketLifecycle(params)
```

Request Parameters

Table 12-12 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 12-13 Responses

Type	Description
Table 12-14 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 12-14 .

Table 12-14 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 12-15 .
InterfaceResult	Table 12-16	Explanation: Results outputted for a successful call. For details, see Table 12-16 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 12-15 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 12-16 GetBucketLifecycleOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server Default value: None
Rules	LifecycleRule []	Yes Explanation: Lifecycle rule information. For details, see Table 12-17 .

Table 12-17 LifecycleRule

Parameter	Type	Description
ID	string	Explanation: Lifecycle rule ID.
Prefix	string	Explanation: Object name prefix. It identifies the objects the rule applies to. You can leave this parameter blank to apply the rule to all objects in the bucket. Assume that you have the following objects: logs/day1 , logs/day2 , logs/day3 , and ExampleObject.jpg . If you set Prefix to ExampleObject.jpg , the rule applies to object ExampleObject.jpg alone. If you set Prefix to logs/ , the rule applies to objects logs/day1 , logs/day2 , and logs/day3 . If you leave Prefix blank, the rule applies to all objects in the bucket.

Parameter	Type	Description
Status	string	Explanation: Whether the rule is enabled. Value range: Enabled Disabled
Transitions	Transition[]	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 12-18 . Restrictions: This parameter is only available for the latest object version.
Expiration	Expiration	Explanation: Object expiration time. For details, see Table 12-19 . Restrictions: This parameter is only available for the latest object version.
NoncurrentVersionTransitions	NoncurrentVersionTransition[]	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 12-20 . Restrictions: • This parameter is only available for historical object versions. • Versioning must be enabled (or suspended after being enabled) for the bucket.
NoncurrentVersionExpiration	NoncurrentVersionExpiration	Explanation: Expiration time of historical object versions. For details, see Table 12-21 . Restrictions: • This parameter is only available for historical object versions. • Versioning must be enabled (or suspended after being enabled) for the bucket. • This parameter is not available for parallel file systems.

Table 12-18 Transition

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Storage class of the object after transition
Date	string	Explanation: OBS executes the lifecycle rule for objects that were modified before the specified date.
Days	number	Explanation: How many days can pass since the last update before the lifecycle rule takes effect

Table 12-19 Expiration

Parameter	Type	Description
Date	string	Explanation: OBS executes the lifecycle rule (deletion) for objects that were modified before the specified date.
Days	number	Explanation: How many days can pass since the last update before the lifecycle rule (deletion) takes effect

Table 12-20 NoncurrentVersionTransition

Parameter	Type	Description
StorageClass	StorageClassType	Explanation: Storage class of historical object versions after transition
NoncurrentDays	number	Explanation: Number of days an object is historical before the specified rule takes effect

Table 12-21 NoncurrentVersionExpiration

Parameter	Type	Description
NoncurrentDays	number	Explanation: Number of days an object is historical before the specified rule takes effect

Table 12-22 AbortIncompleteMultipartUpload

Parameter	Type	Description
DaysAfterInitiation	number	Explanation: Number of days since the initiation of an incomplete multipart upload that OBS will wait before deleting the non-assembled parts (fragments) of the upload

Table 12-23 StorageClassType

Constant	Default Value	Description
ObsClient.enums.StorageClassStandard	STANDARD	Standard storage class. Features low access latency and high throughput and is used for storing massive, frequently accessed (multiple times a month) or small objects (< 1 MB) requiring quick response.
ObsClient.enums.StorageClassWarm	WARM	Infrequent Access storage class. Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but becomes instantly available when needed.
ObsClient.enums.StorageClassCold	COLD	Archive storage class. Used for storing rarely accessed (once a year) data.

Code Examples

You can call **ObsClient.getBucketLifecycle** to view lifecycle rules of a bucket. The following code shows how to obtain the lifecycle rule of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
```



```
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLifecycle() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Obtain the lifecycle rules of the bucket.
    const result = await obsClient.getBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s lifecycle configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      for (let i = 0; i < result.InterfaceResult.Rules.length; i++) {
        const rule = result.InterfaceResult.Rules[i];
        console.log("ID:%s, Prefix:%s, Status:%s, Expiration[Date]:%s, Expiration[Days]:%d,
NoncurrentVersionExpiration[NoncurrentDays]:%s",
          rule.ID, rule.Prefix, rule.Status, rule.Expiration.Date,
          rule.Expiration.Days, rule.NoncurrentVersionExpiration.NoncurrentDays
        );
      };
    }
    return;
  };
  console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
  console.log("Status: %d", result.CommonMsg.Status);
  console.log("Code: %s", result.CommonMsg.Code);
  console.log("Message: %s", result.CommonMsg.Message);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketLifecycle();
```

12.3 Deleting the Lifecycle Rules of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes. For more information, see [Lifecycle Management](#).

This API deletes the lifecycle rules of a bucket.

Restrictions

- To delete the lifecycle rules of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutLifecycleConfiguration** in IAM or **PutLifecycleConfiguration** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.deleteBucketLifecycle(params)
```

Request Parameters

Table 12-24 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 12-25 Responses

Type	Description
Table 12-26 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 12-26 .

Table 12-26 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 12-27 .
InterfaceResult	Table 12-28	Explanation: Results outputted for a successful call. For details, see Table 12-28 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 12-27 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 12-28 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.deleteBucketLifecycle** to delete lifecycle rules of a bucket. The following code shows how to delete the lifecycle rule of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketLifecycle() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Delete the lifecycle configuration of the bucket.
    const result = await obsClient.deleteBucketLifecycle(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s lifecycle configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
  }
}
```

```
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
        console.log("Status: %d", result.CommonMsg.Status);  
        console.log("Code: %s", result.CommonMsg.Code);  
        console.log("Message: %s", result.CommonMsg.Message);  
        console.log("RequestId: %s", result.CommonMsg.RequestId);  
    } catch (error) {  
        console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
        console.log(error);  
    };  
};  
  
deleteBucketLifecycle();
```

13 Bucket CORS (SDK for Node.js)

13.1 Configuring CORS for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Cross-origin resource sharing (CORS) is a mechanism defined by the World Wide Web Consortium (W3C) that allows a web application program in one domain to access resources located in another one. For normal web page requests, website scripts and contents in one domain cannot interact with those in another because of Same Origin Policies (SOPs). OBS supports CORS rules that allow the resources in OBS to be requested by other domains.

This API configures CORS for a bucket. The configured CORS rules follow the principle of new ones overwriting old ones.

Restrictions

- To configure CORS for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketCORS** in IAM or **PutBucketCORS** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketCors(params)
```

Request Parameters

Table 13-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
CorsRules	CorsRule []	Yes	Explanation: List of CORS rules of a bucket. Restrictions: A list can contain a maximum of 100 CORS rules. Value range: See CorsRule . Default value: None

Table 13-2 CorsRule

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	No if used as a request parameter	Explanation: CORS rule ID. Restrictions: None Value range: The value must contain 1 to 255 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
AllowedMethod	string[]	Yes if used as a request parameter	Explanation: The allowed HTTP methods (types of operations on buckets and objects) for a cross-origin request. Restrictions: None Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None
AllowedOrigin	string[]	Yes if used as a request parameter	Explanation: The origin that is allowed to access the bucket. Restrictions: Domain name of the origin. Each origin can contain at most one wildcard character (*). Example: https://*.vbs.example.com Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
AllowedHeader	string[]	No if used as a request parameter	Explanation: The allowed cross-origin request headers. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
MaxAgeSeconds	number	No if used as a request parameter	Explanation: How long the response can be cached on a client Restrictions: Each CORS rule can contain at most one MaxAgeSeconds . Value range: 0 to ($2^{31} - 1$), in seconds Default value: 100

Parameter	Type	Mandatory (Yes/No)	Description
ExposeHeader	string[]	No if used as a request parameter	Explanation: It specifies additional headers a CORS rule allows in a response, which can be used to provide extra information to clients. By default, a browser can access only headers Content-Length and Content-Type . If the browser needs to access other headers, you need to configure them as additional headers. Restrictions: Spaces, asterisks (*), ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None

Responses

Table 13-3 Responses

Type	Description
Table 13-4 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 13-4 .

Table 13-4 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 13-5 .

Parameter	Type	Description
InterfaceResult	Table 13-6	Explanation: Results outputted for a successful call. For details, see Table 13-6 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 13-5 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 13-6 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example configures CORS for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketCors() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify CORS rules.
      CorsRules: [
        {
          // Specify the allowed request methods, which can be GET, PUT, DELETE, POST, or HEAD.
          AllowedMethod: ['GET', 'HEAD', 'PUT'],
          // Specify the allowed request origins.
          AllowedOrigin: ['http://www.a.com', 'http://www.b.com'],
          // Specify whether headers specified in Access-Control-Request-Headers in an OPTIONS preflight
          // request can be used.
          AllowedHeader: ['x-obs-header'],
          // Specify what headers users can access from application programs.
          ExposeHeader: ['x-obs-expose-header'],
          // Specify the browser's cache time of the returned results of OPTIONS preflight requests for specific
          // resources, in seconds.
          MaxAgeSeconds: 10
        }
      ]
    };
    // Configure CORS settings for the bucket.
    const result = await obsClient.setBucketCors(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s) CORS configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
}
```

```
setBucketCors();
```

13.2 Obtaining the CORS Configuration of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Cross-origin resource sharing (CORS) is a mechanism defined by the World Wide Web Consortium (W3C) that allows a web application program in one domain to access resources located in another one. For normal web page requests, website scripts and contents in one domain cannot interact with those in another because of Same Origin Policies (SOPs). OBS supports CORS rules that allow the resources in OBS to be requested by other domains.

This API returns the CORS configuration of a bucket.

Restrictions

- To obtain the CORS configuration of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketCORS** in IAM or **GetBucketCORS** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketCors(params)
```

Request Parameters

Table 13-7 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 13-8 Responses

Type	Description
Table 13-9 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 13-9 .

Table 13-9 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 13-10 .
InterfaceResult	Table 13-11	Explanation: Results outputted for a successful call. For details, see Table 13-11 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 13-10 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 13-11 GetBucketCorsOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
CorsRules	CorsRule[]	Explanation: CORS rule list of the bucket. For details, see CorsRule .

Table 13-12 CorsRule

Parameter	Type	Mandatory (Yes/No)	Description
ID	string	No if used as a request parameter	Explanation: CORS rule ID. Restrictions: None Value range: The value must contain 1 to 255 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
AllowedMethod	string[]	Yes if used as a request parameter	Explanation: The allowed HTTP methods (types of operations on buckets and objects) for a cross-origin request. Restrictions: None Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None
AllowedOrigin	string[]	Yes if used as a request parameter	Explanation: The origin that is allowed to access the bucket. Restrictions: Domain name of the origin. Each origin can contain at most one wildcard character (*). Example: https://*.vbs.example.com Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
AllowedHeader	string[]	No if used as a request parameter	Explanation: The allowed cross-origin request headers. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain at most one wildcard character (*). Spaces, ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None
MaxAgeSeconds	number	No if used as a request parameter	Explanation: How long the response can be cached on a client Restrictions: Each CORS rule can contain at most one MaxAgeSeconds. Value range: 0 to ($2^{31} - 1$), in seconds Default value: 100

Parameter	Type	Mandatory (Yes/No)	Description
ExposeHeader	string[]	No if used as a request parameter	Explanation: It specifies additional headers a CORS rule allows in a response, which can be used to provide extra information to clients. By default, a browser can access only headers Content-Length and Content-Type. If the browser needs to access other headers, you need to configure them as additional headers. Restrictions: Spaces, asterisks (*), ampersands (&), colons (:), less-than signs (<), and full-width characters are not allowed. Value range: None Default value: None

Code Examples

This example returns the CORS configuration of bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketCors() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Obtain the CORS configuration of the bucket.
```

```
const result = await obsClient.getBucketCors(params);
if (result.CommonMsg.Status <= 300) {
  console.log("Get bucket(%s) CORS configuration successful!", params.Bucket);
  console.log("RequestId: %s", result.CommonMsg.RequestId);
  for (let k = 0; k < result.InterfaceResult.CorsRules.length; k++) {
    const corsRule = result.InterfaceResult.CorsRules[k];
    console.log("ID:%s, AllowedOrigin:%s, AllowedMethod:%s, AllowedHeader:%s, MaxAgeSeconds:%d, ExposeHeader:%s", corsRule.ID, corsRule.AllowedOrigin.join("|"), corsRule.AllowedMethod.join("|"), corsRule.AllowedHeader.join("|"), corsRule.MaxAgeSeconds, corsRule.ExposeHeader.join("|"));
  };
  return;
}
console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
getBucketCors();
```

13.3 Deleting the CORS Configuration of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

Cross-origin resource sharing (CORS) is a mechanism defined by the World Wide Web Consortium (W3C) that allows a web application program in one domain to access resources located in another one. For normal web page requests, website scripts and contents in one domain cannot interact with those in another because of Same Origin Policies (SOPs). OBS supports CORS rules that allow the resources in OBS to be requested by other domains.

This API deletes the CORS rules of a bucket.

Restrictions

- To delete the CORS configuration of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketCORS** in IAM or **PutBucketCORS** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.deleteBucketCors(params)
```

Request Parameters

Table 13-13 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.) – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 13-14 Responses

Type	Description
Table 13-15 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 13-15 .

Table 13-15 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 13-16 .
InterfaceResult	Table 13-17	Explanation: Results outputted for a successful call. For details, see Table 13-17 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 13-16 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 13-17 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

Sample code is as follows:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketCors() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Delete CORS rules of the bucket.
    const result = await obsClient.DeleteBucketCors(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s CORS configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
  } catch {
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error");
  }
}
```

```
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("Delete bucket(%s)'s CORS configuration successful!", params.Bucket);
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
deleteBucketCors();
```

14 Logging (SDK for Node.js)

14.1 Configuring Logging for a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

NOTE

A bucket in the Infrequent Access or Archive storage class cannot be used as a log target bucket.

Function

This API enables logging for a bucket (source) and configures another bucket (target) to store the log files. When a bucket is created, logging is not enabled by default. You can call this API to enable logging for the bucket. With logging enabled, a log message is generated for each operation on the bucket. Multiple log messages are packed into a file. The target bucket for storing log files must be specified when logging is enabled. It can be the bucket logging is enabled for, or any other bucket you have access to. If you specify another bucket for storing logs, the bucket must be in the same region as the logged bucket. You can also specify access permissions and name prefixes for log files.

Restrictions

- OBS creates log files and uploads them to the bucket. Before enabling logging for a bucket, you need to create an IAM agency to delegate OBS to upload log files to the specified bucket. For details about how to create an agency, see [Cloud Service Delegation](#).
- To configure logging for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketLogging** in IAM or

PutBucketLogging in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).
- The source bucket and target bucket must be in the same region.
- Before configuring bucket logging, you need to create an agency for OBS on [IAM](#) and obtain the agency name. For details, see [Creating an IAM Agency](#).

Method

```
ObsClient.setBucketLogging(params)
```

Request Parameters

Table 14-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Agency	string	Yes if used in a request for enabling bucket logging	Explanation: Name of the IAM agency created by the owner of the target bucket for OBS. Restrictions: By default, the IAM agency only requires the PutObject permission to upload logs to the target bucket. If default encryption is enabled for the target bucket, the agency also requires the KMS Administrator permission in the region where the target bucket is located. Value range: You can select an existing IAM agency or create one. For details about how to create an agency, see Creating an IAM Agency. Default value: None
LoggingEnabled	Logging Enabled	Yes if you enable logging for the bucket Do not set this parameter when you disable logging for the bucket.	Explanation: Logging configuration information. If this parameter is not set, bucket logging is disabled by default. Restrictions: None Value range: See LoggingEnabled. Default value: None

Table 14-2 LoggingEnabled

Parameter	Type	Mandatory (Yes/No)	Description
TargetBucket	string	Yes if you enable logging for the bucket Do not set this parameter when you disable logging for the bucket.	Explanation: Name of the bucket for storing log files. Restrictions: • This bucket must be in the same region as the bucket with logging enabled. • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
TargetPrefix	string	Yes if you enable logging for the bucket Do not set this parameter when you disable logging for the bucket.	Explanation: Name prefix for log files stored in the target bucket. Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None
TargetGrants	Grant	No	Explanation: Permission information list of grantees, which defines grantees and their permissions for log files. Restrictions: None Value range: See Grant . Default value: None

Table 14-3 Grant

Parameter	Type	Mandatory (Yes/No)	Description
Grantee	Grantee	Yes	Explanation: Grantees and permissions Restrictions: None Value range: See Grantee . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Permission	PermissionType	Yes	Explanation: Granted permission Restrictions: None Value range: See PermissionType . Default value: None

Table 14-4 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
Type	GranteeType	Yes if used as a request parameter	Explanation: Grantee type Restrictions: None Value range: See Table 14-5 . Default value: None
ID	string	Yes if this parameter is used as a request parameter and Type is set to a user	Explanation: Account (domain) ID of the grantee. Restrictions: None Value range: To obtain an account ID, see Obtaining the Account ID . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Name	string	No if used as a request parameter	Explanation: Account name of the grantee. Restrictions: • The account name starts with a letter. • The account name contains 6 to 32 characters. • The account name only allows letters, digits, hyphens (-), and underscores (_). Value range: None Default value: None
URI	Group UriType	Yes if this parameter is used as a request parameter and Type is set to a group	Explanation: Authorized user group Restrictions: None Value range: See Table 14-6 . Default value: None

Table 14-5 GranteeType

Constant	Default Value	Description
GranteeGroup	Group	Grants permissions to user groups.
GranteeUser	CanonicalUser	Grants permissions to individual users.

Table 14-6 GroupUriType

Constant	Default Value	Description
GroupAllUsers	AllUsers	All users
GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.
GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 14-7 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadAcp	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.
ObsClient.enums.PermissionWriteAcp	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Responses

Table 14-8 Responses

Type	Description
Table 14-9 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 14-9 .

Table 14-9 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 14-10 .
InterfaceResult	Table 14-11	Explanation: Results outputted for a successful call. For details, see Table 14-11 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 14-10 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 14-11 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples: Enabling Bucket Logging

This example configures logging for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLogging() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an agency name (obs_test_agency in this example).
      Agency: 'obs_test_agency',
      LoggingEnabled: {
        // Specify a bucket (TargetBucketname in this example) for storing generated log files.
        TargetBucket: 'TargetBucketname',
        // Specify a prefix (TargetPrefixtest/ in this example) for log files to be generated.
        TargetPrefix: 'TargetPrefixtest/',
        // Specify the grantee permissions.
        TargetGrants: [
          // Grant the read permission to a specific user. In this example, the user ID is
          // 0a03f5833900d3730f13c00f49d5exxx.
          { Grantee: { Type: 'CanonicalUser', ID: '0a03f5833900d3730f13c00f49d5exxx' }, Permission:
            obsClient.enums.PermissionRead }
        ]
      }
    };
    // Configure logging for a bucket.
    const result = await obsClient.setBucketLogging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%)s logging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an
    error response.");
    console.log("Status: %d", result.CommonMsg.Status);
  }
}
```

```
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
        console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
        console.log(error);
    };
};
};
setBucketLogging();
```

Code Examples: Disabling Bucket Logging

This example disables logging for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
    //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard coding may result in leakage.
    //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways.
    // security_token: process.env.SECURITY_TOKEN,
    // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here in this example. Replace it with the one currently in use.
    server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketLogging() {
    try {
        const params = {
            // Specify the bucket name.
            Bucket: "examplebucket",
            // Clear the logging configuration.
            LoggingEnabled : {}
        };
        // Delete the access logging configuration of the bucket.
        const result = await obsClient.setBucketLogging(params);
        if (result.CommonMsg.Status <= 300) {
            console.log("Delete bucket(%s)'s logging configuration successful!", params.Bucket);
            console.log("RequestId: %s", result.CommonMsg.RequestId);
            return;
        };
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
        console.log("Status: %d", result.CommonMsg.Status);
        console.log("Code: %s", result.CommonMsg.Code);
        console.log("Message: %s", result.CommonMsg.Message);
        console.log("RequestId: %s", result.CommonMsg.RequestId);
    } catch (error) {
        console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
        console.log(error);
    };
};
setBucketLogging();
```

14.2 Obtaining the Logging Configuration of a Bucket (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

This API returns the logging configuration of a bucket.

Restrictions

- To obtain the logging configuration of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketLogging** in IAM or **GetBucketLogging** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketLogging(params)
```

Request Parameters

Table 14-12 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 14-13 Responses

Type	Description
Table 14-14 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 14-14 .

Table 14-14 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 14-15 .
InterfaceResult	Table 14-16	Explanation: Results outputted for a successful call. For details, see Table 14-16 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 14-15 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 14-16 GetBucketLoggingOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Agency	string	Explanation: Name of the IAM agency created by the owner of the target bucket for OBS. You can select an existing IAM agency or create one. For details about how to create an agency, see Creating an IAM Agency .
LoggingEnabled	LoggingEnabled	Explanation: Logging configuration information

Table 14-17 LoggingEnabled

Parameter	Type	Description
TargetBucket	string	Explanation: Name of the bucket for storing log files.
TargetPrefix	string	Explanation: Name prefix for log files stored in the target bucket.
TargetGrants	Grant	Explanation: Permission information list of grantees, which defines grantees and their permissions for log files. For details, see Table 14-18 .

Table 14-18 Grant

Parameter	Type	Mandatory (Yes/No)	Description
Grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 14-19 .
Permission	PermissionType	Yes if used as a request parameter	Explanation: Granted permissions. For details, see Table 14-22 .

Table 14-19 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
Type	GranteeType	Yes if used as a request parameter	Explanation: Grantee type. For details, see Table 14-20 .
ID	string	Yes if this parameter is used as a request parameter and Type is set to a user	Explanation: Account (domain) ID of the grantee.
Name	string	No if used as a request parameter	Explanation: Account name of the grantee.
URI	GroupUriType	Yes if this parameter is used as a request parameter and Type is set to a group	Explanation: Authorized user group. For details, see Table 14-21 .

Table 14-20 GranteeType

Constant	Default Value	Description
GranteeGroup	Group	Grants permissions to user groups.
GranteeUser	CanonicalUser	Grants permissions to individual users.

Table 14-21 GroupUriType

Constant	Default Value	Description
GroupAllUsers	AllUsers	All users
GroupAuthenticatedUsers	AuthenticatedUsers	Authorized users. This constant is deprecated.
GroupLogDelivery	LogDelivery	Log delivery group. This constant is deprecated.

Table 14-22 PermissionType

Constant	Default Value	Description
ObsClient.enums.PermissionRead	READ	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	WRITE	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadACP	READ_ACP	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default.
ObsClient.enums.PermissionWriteACP	WRITE_ACP	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for their bucket or object by default. This permission allows the grantee to change the access control policies, meaning the grantee has full control over a bucket or object.

Constant	Default Value	Description
ObsClient.enums.PermissionFullControl	FULL_CONTROL	A grantee with this permission for a bucket has PermissionRead , PermissionWrite , PermissionReadAcp , and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead , PermissionReadAcp , and PermissionWriteAcp permissions for the object.

Code Examples

You can call **ObsClient.getBucketLogging** to view the logging configuration of a bucket. The following code shows how to obtain the logging configuration of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketLogging() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Obtain logging for a bucket.
    const result = await obsClient.getBucketLogging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get bucket(%s)'s logging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      if (result.InterfaceResult.LoggingEnabled) {
        console.log('TargetBucket: %s', result.InterfaceResult.LoggingEnabled.TargetBucket);
        console.log('TargetPrefix: %s', result.InterfaceResult.LoggingEnabled.TargetPrefix);
        for (let i = 0; i < result.InterfaceResult.LoggingEnabled.TargetGrants.length; i++) {
          const grant = result.InterfaceResult.LoggingEnabled.TargetGrants[i];
          console.log('Grant[%d]-Type:%s, ID:%s, URI:%s, Permission:%s',
            i, grant.GrantType, grant.GrantID, grant.GrantURI, grant.Permission);
        };
      };
    };
  };
  return;
}
```

```
    };  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an  
error response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  };  
};  
  
getBucketLogging();
```

15 Static Website Hosting (SDK for Node.js)

15.1 Overview (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

You can upload the content files of the static website to your bucket in OBS as objects and configure the **public-read** permission on the files, and then configure the static website hosting mode for your bucket to host your static websites in OBS. After this, when third-party users access your websites, they actually access the objects in your bucket in OBS. When using static website hosting, you can configure request redirection to redirect specific or all requests.

For more information, see [Static Website Hosting](#).

You can perform the following to host a website file in a bucket:

- Step 1** Upload a website file to your bucket in OBS as an object and specify the MIME type for the object.
- Step 2** Set the object ACL to public read.
- Step 3** Access the object using a browser.

----End

Sample code:

```
// Import the OBS library.  
// Use npm for installation.  
const ObsClient = require("esdk-obs-nodejs");  
// Use the source code for installation.  
// var ObsClient = require('./lib/obs');
```

```
// Create an ObsClient instance.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  as an example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. test.html is used in this example.
      Key: 'test.html',
      Body: '<html><header></header><body><h1>Hello OBS</h1></body></html>',
      // Set the MIME type for the object.
      ContentType: 'text/html',
      // Set the object ACL to public read.
      ACL: obsClient.enums.AclPublicRead
    };

    // Upload the object.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!!!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log("StorageClass:%s, ETag:%s", result.InterfaceResult.StorageClass, result.InterfaceResult.ETag);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.")
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.")
    console.log(error);
  }
}

putObject();
```

NOTE

- You can enter **<http://examplebucket.your-endpoint/test.html>** in a browser to access the file hosted using the sample code.
- Online preview is not supported. If online preview is required, see https://support.huaweicloud.com/intl/en-us/usermanual-obs/obs_03_0342.html.

15.2 Configuring Static Website Hosting (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can host static website resources such as HTML web pages, flash files, or audio and video files in an OBS bucket, so that you can provide these hosted resources using the bucket's website endpoint to end users. Typical use cases include:

- Redirecting all requests to another website
- Redirecting specific requests

This API configures static website hosting for a bucket.

Restrictions

- Periods (.) should be avoided in the target bucket name, or there may be certificate verification failures on the client when using HTTPS for access.
- The request body of the website configuration cannot exceed 10 KB.
- To configure static website hosting for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketWebsite** in IAM or **PutBucketWebsite** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketWebsite(params)
```

Request Parameters

Table 15-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
RedirectAllRequestsTo	RedirectAllRequestTo	No	Explanation: Redirection rule for all requests Restrictions: None Value range: See Table 15-2 . Default value: None
IndexDocument	IndexDocument	No	Explanation: Default page configuration. Restrictions: None Value range: See Table 15-3 . Default value: None
ErrorDocument	ErrorDocument	No	Explanation: Error page configuration. Restrictions: None Value range: See Table 15-4 . Default value: None
RoutingRules	RoutingRule[]	No	Explanation: Redirection rule list Restrictions: None Value range: See Table 15-5 . Default value: None

 NOTE

- **ErrorDocument**, **IndexDocument**, and **RoutingRules** must be used together and must not be used with **RedirectAllRequestsTo**.
- When **ErrorDocument**, **IndexDocument**, and **RoutingRules** are used together, **RoutingRules** can be left blank.
- You must set either these three parameters (**ErrorDocument**, **IndexDocument**, and **RoutingRules**) or **RedirectAllRequestsTo**.

Table 15-2 RedirectAllRequestsTo

Parameter	Type	Man dato ry (Yes /No)	Description
HostName	string	Yes if RedirectAllRequestsTo is specified	Explanation: Domain name used for redirection, for example, www.example.com Restrictions: The domain name must comply with the domain name standards. Value range: None Default value: None
Protocol	string	No	Explanation: Protocol used for redirection Restrictions: None Value range: http or https Default value: None

Table 15-3 IndexDocument

Parameter	Type	Mandatory (Yes/No)	Description
Suffix	string	Yes if Index Document is specified	Explanation: Suffix that is appended to the request for a directory. For example, if the suffix is index.html and you request samplebucket/images/ , the returned data will be for the object named images/index.html in the bucket samplebucket . Restrictions: This parameter can neither be left blank nor contain slashes (/). Value range: None Default value: None

Table 15-4 ErrorDocument

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	No if ErrorDocument is specified	Explanation: The object name that is used when a 4XX error occurs. This element specifies the page that is returned when an error occurs. Restrictions: None Value range: The key can contain 1 to 1,024 characters. Default value: None

Table 15-5 RoutingRule

Parameter	Type	Mandatory (Yes/No)	Description
Condition	Condition	No	Explanation: Conditions that must be met for the specified rule to take effect Restrictions: None Value range: See Table 15-6 . Default value: None
Redirect	Redirect	Yes if RoutingRule is specified	Explanation: Details about the redirection Restrictions: None Value range: See Table 15-7 . Default value: None

Table 15-6 Condition

Parameter	Type	Mandatory (Yes/No)	Description
KeyPrefixEquals	string	No	Explanation: Object name prefix for the redirection to take effect. If the name prefix of the requested object is the same as the value specified for this parameter, the redirection rule takes effect. For example, to redirect the requests for the object ExamplePage.html, set KeyPrefixEquals to ExamplePage.html. Restrictions: This parameter cannot be used together with HttpErrorCodeReturnedEquals. Value range: The value can contain 1 to 1,024 characters. Default value: None
HttpErrorCodeReturnedEquals	string	No	Explanation: HTTP error code for the redirection to take effect. If an error code returned is the same as the value specified for this parameter, the redirection rule takes effect. For example, if you want to redirect requests to NotFound.html when HTTP error code 404 is returned, set HttpErrorCodeReturnedEquals to 404 in Condition and set ReplaceKeyWith to NotFound.html in Redirect. Restrictions: This parameter cannot be used together with KeyPrefixEquals. Value range: See Error Codes. Default value: None

Table 15-7 Redirect

Parameter	Type	Mandatory (Yes/No)	Description
Protocol	string	No if used as a request parameter	Explanation: Protocol used for redirection Restrictions: None Value range: http or https Default value: None
HostName	string	No if used as a request parameter	Explanation: Domain name used for redirection Restrictions: The domain name must comply with the domain name standards. Value range: None Default value: None
ReplaceKeyPrefix-With	string	No if used as a request parameter	Explanation: Object name prefix used in the redirection request Restrictions: None Value range: The value can contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
ReplaceKeyWith	string	No if used as a request parameter	Explanation: Object name used in the redirection request Restrictions: This parameter cannot be used together with replaceKeyPrefixWith . Value range: The value can contain 1 to 1,024 characters. Default value: None
HttpRedirectCode	string	No if used as a request parameter	Explanation: HTTP status code in the response to the redirection request Restrictions: None Value range: See Status Codes . Default value: None

Responses

Table 15-8 Responses

Type	Description
Table 15-9 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 15-9 .

Table 15-9 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 15-10 .
InterfaceResult	Table 15-11	Explanation: Results outputted for a successful call. For details, see Table 15-11 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 15-10 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 15-11 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

This example configures website hosting for bucket **examplebucket**.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketWebsite() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify a default page (index.html in this example).
      IndexDocument: { Suffix: 'index.html' },
      // Specify an error page (error.html in this example).
      ErrorDocument: { Key: 'error.html' },
      // Configure redirection for all requests.
      RedirectAllRequestsTo : { HostName : 'www.example.com', Protocol : 'http' }
      // Specify redirect rules for requests.
      RoutingRules: [
        { Redirect: { HostName: "www.a.com", Protocol: obs.ProtocolHttp, ReplaceKeyPrefixWith: "prefix",
          HttpRedirectCode: "304" } },
        { Redirect: { HostName: "www.b.com", Protocol: obs.ProtocolHttps, ReplaceKeyWith: "replaceKey",
          HttpRedirectCode: "304" } },
      ]
    };
    // Configure the website settings for the bucket.
    const result = await obsClient.setBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s website configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
  } catch (error) {
    console.log(error);
  }
}
```

```
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
  console.log("An Exception was found, which means the client encountered an internal problem when
  attempting to communicate with OBS, for example, the client was unable to access the network.");
  console.log(error);
};
};
};

setBucketWebsite();
```

15.3 Obtaining Static Website Hosting Configurations (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can host static website resources such as HTML web pages, flash files, or audio and video files in an OBS bucket, so that you can provide these hosted resources using the bucket's website endpoint to end users. Typical use cases include:

- Redirecting all requests to another website
- Redirecting specific requests

This API returns the static website hosting configurations of the bucket.

Restrictions

- To obtain the static website hosting configurations of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketWebsite** in IAM or **GetBucketWebsite** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketWebsite(params)
```

Request Parameters

Table 15-12 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 15-13 Responses

Type	Description
Table 15-14 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 15-14 .

Table 15-14 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 15-15 .
InterfaceResult	Table 15-16	Explanation: Results outputted for a successful call. For details, see Table 15-16 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 15-15 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 15-16 GetBucketWebsiteOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
RedirectAllRequestsTo	RedirectAllRequestsTo	Explanation: Redirection rules for all requests. For details, see Table 15-19 .
IndexDocument	IndexDocument	Explanation: Default page configuration. For details, see Table 15-17 .
ErrorDocument	ErrorDocument	Explanation: Error page configuration. For details, see Table 15-18 .
RoutingRules	RoutingRule []	Explanation: List of routing rules. For details, see Table 15-20 .

Table 15-17 IndexDocument

Parameter	Type	Mandatory (Yes/No)	Description
Suffix	string	Yes if Index Document is specified	Explanation: Suffix that is appended to the request for a directory. For example, if the suffix is index.html and you request samplebucket/images/ , the returned data will be for the object named images/index.html in the bucket samplebucket . Restrictions: This parameter can neither be left blank nor contain slashes (/). Value range: None Default value: None

Table 15-18 ErrorDocument

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	No if ErrorDocument is specified	Explanation: The object name that is used when a 4XX error occurs. This element specifies the page that is returned when an error occurs. Restrictions: None Value range: The key can contain 1 to 1,024 characters. Default value: None

Table 15-19 RedirectAllRequestsTo

Parameter	Type	Description
HostName	string	Explanation: Domain name used for redirection, for example, www.example.com
Protocol	string	Explanation: Protocol used for redirection

Table 15-20 RoutingRule

Parameter	Type	Description
Condition	Condition	Explanation: Conditions that must be met for the specified rule to take effect. For details, see Condition .
Redirect	Redirect	Explanation: Details about the redirection. For details, see Table 15-22 .

Table 15-21 Condition

Parameter	Type	Mandatory (Yes/No)	Description
KeyPrefixEquals	string	No	Explanation: Object name prefix for the redirection to take effect. If the name prefix of the requested object is the same as the value specified for this parameter, the redirection rule takes effect. For example, to redirect the requests for the object ExamplePage.html, set KeyPrefixEquals to ExamplePage.html. Restrictions: This parameter cannot be used together with HttpErrorCodeReturnedEquals. Value range: The value can contain 1 to 1,024 characters. Default value: None
HttpErrorCodeReturnedEquals	string	No	Explanation: HTTP error code for the redirection to take effect. If an error code returned is the same as the value specified for this parameter, the redirection rule takes effect. For example, if you want to redirect requests to NotFound.html when HTTP error code 404 is returned, set HttpErrorCodeReturnedEquals to 404 in Condition and set ReplaceKeyWith to NotFound.html in Redirect. Restrictions: This parameter cannot be used together with KeyPrefixEquals. Value range: See Error Codes. Default value: None

Table 15-22 Redirect

Parameter	Type	Description
Protocol	string	Explanation: Protocol used for redirection
HostName	string	Explanation: Domain name used for redirection
ReplaceKeyPrefixWith	string	Explanation: Object name prefix used in the redirection request
ReplaceKeyWith	string	Explanation: Object name used in the redirection request
HttpRedirectCode	string	Explanation: HTTP status code in the response to the redirection request. For details, see Status Codes .

Code Examples

You can call **ObsClient.getBucketWebsite** to obtain the hosting configuration of a bucket. The following code shows how to obtain the website configuration of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getBucketWebsite() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Obtain the website configuration of the bucket.
    const result = await obsClient.getBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
```

```
console.log("Get bucket(%s)'s website configuration successful!", params.Bucket);
console.log('RedirectAllRequestsTo:');
console.log('HostName: %s', result.InterfaceResult.RedirectAllRequestsTo.HostName);
console.log('Protocol: %s', result.InterfaceResult.RedirectAllRequestsTo.Protocol);
console.log('IndexDocument[Suffix]: %s', result.InterfaceResult.IndexDocument.Suffix);
console.log('ErrorDocument[Key]: %s', result.InterfaceResult.ErrorDocument.Key);
console.log('RoutingRules:');
for (let i = 0; i < result.InterfaceResult.RoutingRules; i++) {
    let routingRule = result.InterfaceResult.RoutingRules[i];
    let condition = routingRule.Condition;
    console.log("Condition[%d]-KeyPrefixEquals:%s, HttpStatusCodeReturnedEquals:%s",
        i, condition.KeyPrefixEquals, condition.HttpStatusCodeReturnedEquals
    );
    let redirect = routingRule.Redirect;
    console.log("Redirect[%d]-Protocol:%s, HostName:%s, ReplaceKeyPrefixWith:%s,
HttpRedirectCode:%s",
        i, redirect.Protocol, redirect.HostName, redirect.ReplaceKeyPrefixWith, redirect.HttpRedirectCode
    );
};
return;
};

console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
console.log("Status: %d", result.CommonMsg.Status);
console.log("Code: %s", result.CommonMsg.Code);
console.log("Message: %s", result.CommonMsg.Message);
console.log("RequestId: %s", result.CommonMsg.RequestId);
} catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
};
};
};

getBucketWebsite();
```

15.4 Deleting Website Hosting Settings (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

You can host static website resources such as HTML web pages, flash files, or audio and video files in an OBS bucket, so that you can provide these hosted resources using the bucket's website endpoint to end users. Typical use cases include:

- Redirecting all requests to another website
- Redirecting specific requests

This API deletes the static website hosting configurations of a bucket.

Restrictions

- To delete the static website hosting configurations of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:DeleteBucketWebsite** in IAM or **DeleteBucketWebsite** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.deleteBucketWebsite(params)
```

Request Parameters

Table 15-23 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 15-24 Responses

Type	Description
Table 15-25 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 15-25 .

Table 15-25 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 15-26 .
InterfaceResult	Table 15-27	Explanation: Results outputted for a successful call. For details, see Table 15-27 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 15-26 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 15-27 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.deleteBucketWebsite** to delete the hosting configuration of a bucket. The following code shows how to delete the website configuration of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs')
;
// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketWebsite() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
    };
    // Delete the website configuration of the bucket.
    const result = await obsClient.deleteBucketWebsite(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s website configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
  }
}
```



```
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
        console.log("Status: %d", result.CommonMsg.Status);  
        console.log("Code: %s", result.CommonMsg.Code);  
        console.log("Message: %s", result.CommonMsg.Message);  
        console.log("RequestId: %s", result.CommonMsg.RequestId);  
    } catch (error) {  
        console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
        console.log(error);  
    };  
};  
deleteBucketWebsite();
```

16 Tagging (SDK for Node.js)

16.1 Setting Bucket Tags (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

If you add tags to a bucket, SDRs generated for the requests sent to this bucket will include these tags, so you can use the tags to classify SDRs for detailed cost analysis. For example, if you have an application that uploads its data to a bucket when it is running, you can tag the bucket with the name of this application. Then, you can analyze the cost of this application by using that tag.

This API adds tags to a bucket.

For more information about bucket tags, see [Tags](#).

Restrictions

- A bucket can have a maximum of 10 tags.
- A tag key and its value can contain a maximum of 128 and 255 characters, respectively.
- Tag keys and values cannot contain commas (,), asterisks (*), vertical bars (|), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII codes (0x00 to 0x1F).
- To add tags to a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketTagging** in IAM or **PutBucketTagging** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.setBucketTagging(params)
```

Request Parameters

Table 16-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Bucket name Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Tags	Tag []	Yes	Explanation: Bucket tag set. Restrictions: • A bucket can have a maximum of 10 tags. Each tag can contain only one key-value pair. • For the same bucket, tag keys must be unique, but tag values can be duplicated or left blank. Value range: See Tag . Default value: None

Table 16-2 Tag

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes if used as a request parameter	Explanation: Tag key. Restrictions: • The tag key in the same bucket must be unique. • You can define tags or select the ones predefined on TMS. • The key must contain 1 to 128 characters. • Tag keys cannot start or end with a space and cannot contain commas (,), asterisks (*), vertical bars (), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII control characters (0x00 to 0x1F). Tag keys and values must be URL encoded before being sent to a server. • The key is case-sensitive. Value range: The key must contain 1 to 128 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Value	string	Yes if used as a request parameter	Explanation: Tag value. Restrictions: Tag values can be duplicated or left blank. • The value must contain 0 to 255 characters. Tag values cannot start or end with a space and cannot contain commas (,), asterisks (*), vertical bars (), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII control characters (0x00 to 0x1F). Tag keys and values must be URL encoded before being sent to a server. • The value is case-sensitive. Value range: The value must contain 0 to 255 characters. Default value: None

Responses

Table 16-3 Responses

Type	Description
Table 16-4 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 16-4 .

Table 16-4 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 16-5 .
InterfaceResult	Table 16-6	Explanation: Results outputted for a successful call. For details, see Table 16-6 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 16-5 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.
RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 16-6 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

The following code shows how to set tags for the **examplebucket** bucket:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function setBucketTagging() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify bucket tags.
      Tags: [
        {Key: "key0", Value: "value0"},
        {Key: "key1", Value: "value1"},
      ]
    };
    // Set bucket tags.
    const result = await obsClient.setBucketTagging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Set bucket(%s)'s tagging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    };
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
    response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
    attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  };
};
```



```
setBucketTagging();
```

16.2 Obtaining Bucket Tags (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

If you add tags to a bucket, SDRs generated for the requests sent to this bucket will include these tags, so you can use the tags to classify SDRs for detailed cost analysis. For example, if you have an application that uploads its data to a bucket when it is running, you can tag the bucket with the name of this application. Then, you can analyze the cost of this application by using that tag.

This API returns the tags of a bucket.

For more information about bucket tags, see [Tags](#).

Restrictions

- To obtain the bucket tags, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketTagging** in IAM or **GetBucketTagging** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).
- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

```
ObsClient.getBucketTagging(params)
```

Request Parameters

Table 16-7 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 16-8 Responses

Type	Description
Table 16-9 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 16-9 .

Table 16-9 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 16-10 .
InterfaceResult	Table 16-11	Explanation: Results outputted for a successful call. For details, see Table 16-11 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 16-10 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 16-11 GetBucketTaggingOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server
Tags	Tag []	Explanation: Bucket tag list. For details, see Table 16-12 .

Table 16-12 Tag

Parameter	Type	Mandatory (Yes/No)	Description
Key	string	Yes if used as a request parameter	Explanation: Tag key. Restrictions: • The tag key in the same bucket must be unique. • You can define tags or select the ones predefined on TMS. • The key must contain 1 to 128 characters. • Tag keys cannot start or end with a space and cannot contain commas (,), asterisks (*), vertical bars (), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII control characters (0x00 to 0x1F). Tag keys and values must be URL encoded before being sent to a server. • The key is case-sensitive. Value range: The key must contain 1 to 128 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
Value	string	Yes if used as a request parameter	Explanation: Tag value. Restrictions: Tag values can be duplicated or left blank. - The value must contain 0 to 255 characters. Tag values cannot start or end with a space and cannot contain commas (,), asterisks (*), vertical bars (), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII control characters (0x00 to 0x1F). Tag keys and values must be URL encoded before being sent to a server. - The value is case-sensitive. Value range: The value must contain 0 to 255 characters. Default value: None

Code Examples

You can call **ObsClient.getBucketTagging** to view bucket tags. The following code shows how to obtain the tag configuration of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  //coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  // use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  // variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  // in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
```

```
});  
  
async function getBucketTagging() {  
  try {  
    const params = {  
      // Specify the bucket name.  
      Bucket: "examplebucket"  
    };  
    // Obtain the bucket tag configuration.  
    const result = await obsClient.getBucketTagging(params);  
    if (result.CommonMsg.Status <= 300) {  
      console.log("Get bucket(%s)'s tagging configuration successful!", params.Bucket);  
      console.log("RequestId: %s", result.CommonMsg.RequestId);  
      for (let tag of result.InterfaceResult.Tags) {  
        console.log('Key: %s, Value: %s', tag.key, tag.value);  
      };  
      return;  
    };  
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");  
    console.log("Status: %d", result.CommonMsg.Status);  
    console.log("Code: %s", result.CommonMsg.Code);  
    console.log("Message: %s", result.CommonMsg.Message);  
    console.log("RequestId: %s", result.CommonMsg.RequestId);  
  } catch (error) {  
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");  
    console.log(error);  
  };  
};  
getBucketTagging();
```

16.3 Deleting Bucket Tags (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Function

If you add tags to a bucket, SDRs generated for the requests sent to this bucket will include these tags, so you can use the tags to classify SDRs for detailed cost analysis. For example, if you have an application that uploads its data to a bucket when it is running, you can tag the bucket with the name of this application. Then, you can analyze the cost of this application by using that tag.

This API deletes the tags of a bucket.

For more information about bucket tags, see [Tags](#).

Restrictions

- To delete bucket tags, you must be the bucket owner or have the required permission (**obs:bucket:DeleteBucketTagging** in IAM or **DeleteBucketTagging** in a bucket policy). For details, see [Introduction to OBS Access Control](#), [IAM Custom Policies](#), and [Creating a Custom Bucket Policy](#).

- To learn about the mappings between OBS regions and endpoints, see [Regions and Endpoints](#).

Method

ObsClient.deleteBucketTagging(params)

Request Parameters

Table 16-13 BucketRequestInput

Parameter	Type	Mandatory (Yes/No)	Description
Bucket	string	Yes	Explanation: Bucket name. Restrictions: • A bucket name must be unique across all accounts and regions. • A bucket name: – Must be 3 to 63 characters long and start with a digit or letter. Lowercase letters, digits, hyphens (-), and periods (.) are allowed. – Cannot be formatted as an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain a period (.) and a hyphen (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets with the same name in the same region, no error will be reported, and the bucket attributes comply with those set in the first creation request. Value range: The value can contain 3 to 63 characters. Default value: None

Responses

Table 16-14 Responses

Type	Description
Table 16-15 NOTE This API returns a Promise response, which requires the Promise or async/await syntax.	Explanation: Returned results. For details, see Table 16-15 .

Table 16-15 Response

Parameter	Type	Description
CommonMsg	ICommonMsg	Explanation: Common information generated after an API call is complete, including the HTTP status code and error code. For details, see Table 16-16 .
InterfaceResult	Table 16-17	Explanation: Results outputted for a successful call. For details, see Table 16-17 . Restrictions: This parameter is not included if the value of Status is greater than 300.

Table 16-16 ICommonMsg

Parameter	Type	Description
Status	number	Explanation: HTTP status code returned by the OBS server. Value range: A status code is a group of digits indicating the status of a response. It ranges from 2xx (indicating successes) to 4xx or 5xx (indicating errors). For details, see Status Codes .
Code	string	Explanation: Error code returned by the OBS server.
Message	string	Explanation: Error description returned by the OBS server.
HostId	string	Explanation: Request server ID returned by the OBS server.

RequestId	string	Explanation: Request ID returned by the OBS server.
Id2	string	Explanation: Request ID2 returned by the OBS server.
Indicator	string	Explanation: Error code details returned by the OBS server.

Table 16-17 BaseResponseOutput

Parameter	Type	Description
RequestId	string	Explanation: Request ID returned by the OBS server

Code Examples

You can call **ObsClient.deleteBucketTagging** to delete bucket tags. The following code shows how to delete the tags of bucket **examplebucket**:

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
  // security_token: process.env.SECURITY_TOKEN,
  // Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
  in this example. Replace it with the one currently in use.
  server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function deleteBucketTagging() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket"
    };
    // Delete the bucket tag configuration.
    const result = await obsClient.deleteBucketTagging(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Delete bucket(%s)'s tagging configuration successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("Delete bucket(%s)'s tagging configuration fail!", params.Bucket);
  }
}
```

```
        console.log("An ObsError was found, which means your request sent to OBS was rejected with an error  
response.");  
        console.log("Status: %d", result.CommonMsg.Status);  
        console.log("Code: %s", result.CommonMsg.Code);  
        console.log("Message: %s", result.CommonMsg.Message);  
        console.log("RequestId: %s", result.CommonMsg.RequestId);  
    } catch (error) {  
        console.log("An Exception was found, which means the client encountered an internal problem when  
attempting to communicate with OBS, for example, the client was unable to access the network.");  
        console.log(error);  
    };  
};  
  
deleteBucketTagging();
```

17 Server-Side Encryption (SDK for Node.js)

17.1 Overview (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

OBS supports server-side encryption.

For more information, see [Server-Side Encryption](#).

17.2 APIs (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

The following table lists APIs related to server-side encryption:

Method in OBS Node.js SDK	Description	Supported Encryption Type
ObsClient.putObject	Sets the encryption algorithm and key during object upload to enable server-side encryption.	SSE-KMS SSE-C

Method in OBS Node.js SDK	Description	Supported Encryption Type
ObsClient.appendObject	Sets the encryption algorithm and key during appendable upload to enable server-side encryption.	SSE-KMS SSE-C
ObsClient.getObject	Sets the decryption algorithm and key during object download to decrypt the object.	SSE-C
ObsClient.copyObject	1. Sets the decryption algorithm and key for decrypting the source object during object copy. 2. Sets the encryption algorithm and key during object copy to enable the encryption algorithm for the target object.	SSE-KMS SSE-C
ObsClient.getObjectMetadata	Sets the decryption algorithm and key when obtaining the object metadata to decrypt the object.	SSE-C
ObsClient.initiateMultipartUpload	Sets the encryption algorithm and key when initializing a multipart upload to enable server-side encryption for the final object generated.	SSE-KMS SSE-C
ObsClient.uploadPart	Sets the encryption algorithm and key during multipart upload to enable server-side encryption for parts.	SSE-C
ObsClient.copyPart	1. Sets the decryption algorithm and key for decrypting the source object during partial object copy. 2. Sets the encryption algorithm and key during partial object copy to enable the encryption algorithm for the target object part.	SSE-C

OBS Node.js SDK supports the following two types of encryption/decryption mode:

Encryption / Decryption Type	Request Parameter	Description
SSE-KMS	SseKms	Indicates that SSE-KMS mode is used. Currently, only kms is supported.

Encryption / Decryption Type	Request Parameter	Description
	SseKmsKey	Indicates the master key used in SSE-KMS mode. The value can be null.
SSE-C	SseC	Indicates that SSE-C mode is used. Currently, only AES256 is supported.
	SseCKey	Indicates the key in SSE-C mode. It is calculated using the AES256 algorithm. This parameter can be used to encrypt an object to be uploaded and decrypt an object to be downloaded.
	CopySourceSseC	Indicates the source object decrypted in SSE-C mode. The value can only be AES256. This parameter is applicable to ObsClient.copyObject and ObsClient.copyPart .
	CopySourceSseCKey	Indicates the key used by a source object for decryption in SSE-C mode. It is calculated using the AES256 algorithm. This parameter is applicable to ObsClient.copyObject and ObsClient.copyPart .

17.3 Code Examples (SDK for Node.js)

NOTICE

If you have any questions during development, post them on the [Issues](#) page of GitHub.

Object Encryption and Upload

This example uploads object **example/objectname** to bucket **examplebucket** and encrypts it using SSE-C.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  // coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/
```

```
intl/en-us/usermanual-ca/ca_01_0003.html.
access_key_id: process.env.ACCESS_KEY_ID,
secret_access_key: process.env.SECRET_ACCESS_KEY,
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
variables or import an AK/SK pair in other ways.
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function putObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify an object. example/objectname is used in this example.
      Key: "example/objectname",
      // Specify a text object.
      Body: 'Hello OBS',
      // Specify SSE-C as the encryption algorithm.
      SseC: 'AES256',
      SseCKey: 'your sse-c key generated by AES-256 algorithm'
    };
    // Upload and encrypt the object.
    const result = await obsClient.putObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Put object(%s) under the bucket(%s) successful!", params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error
response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when
attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};
};
putObject();
```

Object Decryption and Download

This example downloads the encrypted object **example/objectname** using a stream.

```
// Import the OBS library.
// Use npm to install the client.
const ObsClient = require("esdk-obs-nodejs");
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  // Obtain an AK/SK pair using environment variables or import an AK/SK pair in other ways. Using hard
  coding may result in leakage.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  // (Optional) If you use a temporary AK/SK pair and a security token to access OBS, you are advised not to
  use hard coding, which may result in information leakage. You can obtain an AK/SK pair using environment
  variables or import an AK/SK pair in other ways.
```



```
// security_token: process.env.SECURITY_TOKEN,
// Enter the endpoint corresponding to the region where the bucket is located. CN-Hong Kong is used here
// in this example. Replace it with the one currently in use.
server: "https://obs.ap-southeast-1.myhuaweicloud.com"
});

async function getObject() {
  try {
    const params = {
      // Specify the bucket name.
      Bucket: "examplebucket",
      // Specify the object (example/objectname in this example).
      Key: 'example/objectname',
      // Specify SSE-C as the algorithm.
      SseC: 'AES256',
      // Specify the same key you used for encrypting the object during the upload.
      SseCKey: 'your sse-c key generated by AES-256 algorithm'
    };

    // Download the encrypted object.
    const result = await obsClient.getObject(params);
    if (result.CommonMsg.Status <= 300) {
      console.log("Get object(%s) under the bucket(%s) successful!", params.Key, params.Bucket);
      console.log("RequestId: %s", result.CommonMsg.RequestId);
      console.log('Object Content: %s', result.InterfaceResult.Content);
      return;
    }
    console.log("An ObsError was found, which means your request sent to OBS was rejected with an error response.");
    console.log("Status: %d", result.CommonMsg.Status);
    console.log("Code: %s", result.CommonMsg.Code);
    console.log("Message: %s", result.CommonMsg.Message);
    console.log("RequestId: %s", result.CommonMsg.RequestId);
  } catch (error) {
    console.log("An Exception was found, which means the client encountered an internal problem when attempting to communicate with OBS, for example, the client was unable to access the network.");
    console.log(error);
  }
};

getObject();
```

18 Troubleshooting (SDK for Node.js)

18.1 OBS Server-Side Error Codes (SDK for Node.js)

If the OBS server encounters an error when processing a request, a response containing the error code and error description is returned. The following table lists details about each error code and HTTP status code.

Error Code	Description	HTTP Status Code
AccessDenied	Access denied.	403 Forbidden
AccessForbidden	Insufficient permission.	403 Forbidden
AccountProblem	Your account is abnormal (for example, it has been expired or frozen).	403 Forbidden
AllAccessDisabled	You have no permission to perform the operation.	403 Forbidden
AmbiguousGrantByEmailAddress	The provided email address is associated with more than one account.	400 Bad Request
BadDigest	The specified value of Content-MD5 does not match the value received by OBS.	400 Bad Request
BadDomainName	Invalid domain name.	400 Bad Request
BadRequest	Invalid request parameters.	400 Bad Request

Error Code	Description	HTTP Status Code
BucketAlreadyExists	The requested bucket name already exists. The bucket namespace is shared by all users of OBS. Select another name and retry.	409 Conflict
BucketAlreadyOwnedByYou	Your previous request for creating the named bucket succeeded and you already own it.	409 Conflict
BucketNotEmpty	The bucket that you tried to delete is not empty.	409 Conflict
CredentialsNotSupported	This request does not support security credentials.	400 Bad Request
CustomDomainAlreadyExist	The configured domain already exists.	400 Bad Request
CustomDomainNotExist	The domain to be operated does not exist.	400 Bad Request
DeregisterUserId	The user has been deregistered.	403 Forbidden
EntityTooSmall	The size of the object to be uploaded is smaller than the lower limit.	400 Bad Request
EntityTooLarge	The size of the object to be uploaded has exceeded the upper limit.	400 Bad Request
FrozenUserId	The user has been frozen.	403 Forbidden
IllegalVersioningConfigurationException	Invalid versioning configuration in the request.	400 Bad Request
IllegalLocationConstraintException	The configured region limitation is inconsistent with the region where it resides.	400 Bad Request
InArrearOrInsufficientBalance	The user has no permission to perform some operations due to being in arrears or insufficient funds.	403 Forbidden

Error Code	Description	HTTP Status Code
IncompleteBody	Incomplete request body.	400 Bad Request
IncorrectNumberOfFilesInPostRequest	Each POST request must contain one file to be uploaded.	400 Bad Request
InlineDataTooLarge	The size of inline data has exceeded the upper limit.	400 Bad Request
InsufficientStorageSpace	Insufficient storage space.	403 Forbidden
InternalServerError	An internal error occurs. Retry later.	500 Internal Server Error
InvalidAccessKeyId	The access key ID provided by the customer does not exist in the system.	403 Forbidden
InvalidAddressingHeader	The anonymous role must be specified.	N/A
InvalidArgument	Invalid parameter.	400 Bad Request
InvalidBucketName	The specified bucket name in the request is invalid.	400 Bad Request
InvalidBucket	The bucket to be accessed does not exist.	400 Bad Request
InvalidBucketState	Invalid bucket status.	409 Conflict
InvalidBucketStoragePolicy	An invalid new policy is specified during bucket policy modification.	400 Bad Request
InvalidDigest	The specified Content-MD5 in the HTTP header is invalid.	400 Bad Request
InvalidEncryptionAlgorithmError	Incorrect encryption algorithm.	400 Bad Request
InvalidLocationConstraint	The location specified during bucket creation is invalid.	400 Bad Request
InvalidPart	One or more specified parts are not found. The parts may not be uploaded or the specified entity tags (ETags) do not match the parts' ETags.	400 Bad Request

Error Code	Description	HTTP Status Code
InvalidPartOrder	Parts are not listed in ascending order by part number.	400 Bad Request
InvalidPayer	All accesses to this object are disabled.	403 Forbidden
InvalidPolicyDocument	The content of the form does not meet the conditions specified in the policy document.	400 Bad Request
InvalidRange	The requested range cannot be obtained.	416 Client Requested Range Not Satisfiable
InvalidRedirectLocation	Invalid redirect location.	400 Bad Request
InvalidRequest	Invalid request.	400 Bad Request
InvalidRequestBody	Invalid POST request body.	400 Bad Request
InvalidSecurity	Invalid security credentials.	403 Forbidden
InvalidStorageClass	Invalid storage class.	400 Bad Request
InvalidTargetBucketForLogging	The delivery group has no ACL permission for the target bucket.	400 Bad Request
InvalidURI	The specified URI cannot be resolved.	400 Bad Request
KeyTooLong	The provided key is too long.	400 Bad Request
MalformedACLError	The provided XML file has syntax errors or does not meet the format requirements.	400 Bad Request
MalformedError	The XML format in the request is incorrect.	400 Bad Request
MalformedLoggingStatus	The XML format of Logging is incorrect.	400 Bad Request
MalformedPolicy	The bucket policy failed the check.	400 Bad Request

Error Code	Description	HTTP Status Code
MalformedPOSTRequest	The body of the POST request is in an incorrect format.	400 Bad Request
MalformedQuotaError	The Quota XML format is incorrect.	400 Bad Request
MalformedXML	This error code is returned after you send an XML file in incorrect format, stating "The XML you provided was not well-formed or did not validate against our published schema."	400 Bad Request
MaxMessageLengthExceeded	The request is too long.	400 Bad Request
MaxPostPreDataLengthExceeded Error	The POST request fields prior to the file to be uploaded are too large.	400 Bad Request
MetadataTooLarge	The size of the metadata header has exceeded the upper limit.	400 Bad Request
MethodNotAllowed	The specified method is not allowed against the requested resource. The message "Specified method is not supported." is returned.	405 Method Not Allowed
MissingContentLength	The HTTP header Content-Length is not provided.	411 Length Required
MissingRegion	No region contained in the request and no default region defined in the system.	400 Bad Request
MissingRequestBodyError	This error code is returned after you send an empty XML file, stating "Request body is empty."	400 Bad Request
MissingRequiredHeader	Required headers missing in the request.	400 Bad Request
MissingSecurityHeader	A required header is not provided.	400 Bad Request

Error Code	Description	HTTP Status Code
NoSuchBucket	The specified bucket does not exist.	404 Not Found
NoSuchBucketPolicy	No bucket policy exists.	404 Not Found
NoSuchCORSConfiguration	No CORS configuration exists.	404 Not Found
NoSuchCustomDomain	The requested user domain does not exist.	404 Not Found
NoSuchKey	The specified key does not exist.	404 Not Found
NoSuchLifecycleConfiguration	The requested Lifecycle does not exist.	404 Not Found
NoSuchPolicy	The specified policy name does not exist.	404 Not Found
NoSuchUpload	The specified multipart upload does not exist. The upload ID does not exist or the multipart upload has been aborted or completed.	404 Not Found
NoSuchVersion	The specified version ID does not match any existing version.	404 Not Found
NoSuchWebsiteConfiguration	The requested website does not exist.	404 Not Found
NotImplemented	The provided header implies a function that is unavailable.	501 Not Implemented
NotSignedUp	Your account has not registered in the system.	403 Forbidden
OperationAborted	A conflicting operation is being performed on this resource. Retry later.	409 Conflict
PermanentRedirect	The requested bucket has been permanently redirected to a new URL. All future requests must be sent to the new URL.	301 Moved Permanently
PreconditionFailed	At least one of the specified preconditions is not met.	412 Precondition Failed

Error Code	Description	HTTP Status Code
Redirect	The request is temporarily redirected.	307 Moved Temporarily
RequestIsNotMultiPartContent	A bucket POST request must contain an enclosure-type multipart or the form-data.	400 Bad Request
RequestTimeout	The socket connection to the server has no read or write operations within the timeout period.	400 Bad Request
RequestTimeTooSkewed	The request time and the server's time differ a lot.	403 Forbidden
RequestTorrentOfBucketError	Requesting the bucket's torrent file is not allowed.	400 Bad Request
ServiceNotImplemented	The request method is not implemented by the server.	501 Not Implemented
ServiceNotSupported	The request method is not supported by the server.	409 Conflict
ServiceUnavailable	The server is overloaded or has internal errors.	503 Service Unavailable
SignatureDoesNotMatch	The provided signature does not match the signature calculated by OBS. Check your AK/SK and signature calculation method.	403 Forbidden
SlowDown	Too frequent requests. Reduce your request frequency.	503 Service Unavailable
System Capacity Not enough	Insufficient system space.	403 Forbidden
TooManyCustomDomains	Too many user domains are configured.	400 Bad Request
TemporaryRedirect	The request is redirected to the bucket while the domain name server (DNS) is being updated.	307 Moved Temporarily
TooManyBuckets	You have attempted to create more buckets than allowed.	400 Bad Request

Error Code	Description	HTTP Status Code
TooManyObjectCopied	An object has been copied too many times.	400 Bad Request
TooManyWrongSignature	The request is rejected due to high-frequency errors.	400 Bad Request
UnexpectedContent	This request does not support content.	400 Bad Request
UnresolvableGrantByEmailAddress	The provided email address does not match any recorded accounts.	400 Bad Request
UserKeyMustBeSpecified	The user's AK is not carried in the request.	400 Bad Request
WebsiteRedirect	The website request lacks bucketName .	301 Moved Permanently
KMS.DisabledException	The master key is disabled in server-side encryption with KMS-managed keys (SSE-KMS) mode.	400 Bad Request
KMS.NotFoundException	The master key does not exist in SSE-KMS mode.	400 Bad Request
RestoreAlreadyInProgress	The objects are being restored. The request conflicts with another one.	409 Conflict
ObjectHasAlreadyRestored	The objects have been restored and the retention period of the objects cannot be shortened.	409 Conflict
InvalidObjectState	The restored object is not an Archive object.	403 Forbidden
InvalidTagError	An invalid tag is provided when configuring bucket tags.	400 Bad Request
NoSuchTagSet	The specified bucket is not configured with a tag.	404 Not Found

18.2 SDK Common Results (SDK for Node.js)

After you call an API in an instance of the **ObsClient** class, a common result object will be returned if no exception is thrown. The following table lists the fields of the object:

Parameter		Type	Description
CommonMsg		Object	Common information generated after the API is called, including HTTP status code and error code.
-	Status	Number	HTTP status code. If the value is smaller than 300 , the operation succeeds. Otherwise, the operation fails.
	Code	String	Error code returned by the OBS server. If Status is smaller than 300 , the value is null.
	Message	String	Error description returned by the OBS server. If Status is smaller than 300 , the value is null.
	HostId	String	Requested Server ID. If Status is smaller than 300 , the value is null.
	RequestId	String	Request ID returned by the OBS server
	Id2	String	Request ID2 returned by the OBS server
	Indicator	String	Detailed error code returned by the OBS server. If Status is smaller than 300 , the value is null.
InterfaceResult		Object	Result data generated after the operation is successful. If Status is greater than 300 , the value is null.
-	RequestId	String	Request ID returned by the OBS server.
	Id2	String	Request ID2 returned by the OBS server
	Other fields	For details, see the Response area in the corresponding section.	

Sample code:

```
// Import the OBS library.
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an instance of ObsClient.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  //CN-Hong Kong is used here as an example. Replace it with the one in your actual situation.
  server: 'https://obs.ap-southeast-1.myhuaweicloud.com'
});
```

```
// Call APIs to perform operations, such as downloading an object.
obsClient.getObject({
  Bucket : 'bucketname',
  Key : 'objectname',
}, (err, result) => {
  if(!err){
    if(result.CommonMsg.Status < 300){
      // Obtain RequestId.
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      // Obtain other parameters.
      console.log('Content-->' + result.InterfaceResult.Content.toString());
    }else{
      // Obtain Code and Message.
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

18.3 Log Analysis (SDK for Node.js)

Log Configuration

OBS Node.js SDK provides the logging function based on Log4js. You can call **ObsClient.initLog** to enable and configure logging. The sample code is as follows:

```
obsClient.initLog({
  name: 'test', // Log name
  file_full_path: './logs/OBS-SDK.log', //Set the path to the log file.
  max_log_size: 20480, //Set the size of the log file, in bytes.
  backups: 10, //Set the maximum number of log files that can be stored.
  level: 'warn', //Set the log level.
  log_to_console: true //Set whether to print the log to Console.
});
```

NOTE

- The logging function is disabled by default. You need to enable it if needed.
- Use the **file_full_path** parameter to specify the path to the log file. The path can be set to an absolute path or a relative path.

Log Format

The SDK log format is: Log time|log level|invoked interface|log content. The following are examples:

```
2017/10/12 10:21:05 666|INFO |ListBuckets|enter ListBuckets...
2017/10/12 10:21:05 672|INFO |ListBuckets|prepare request parameters ok,then Send request to service start
2017/10/12 10:21:05 715|INFO |ListBuckets|2017-10-12 10:21:05|http cost 34 ms|0|
2017/10/12 10:21:05 716|INFO |ListBuckets|get response start, statusCode:200
```

Log Level

When current logs cannot be used to troubleshoot system faults, you can change the log level to obtain more information. You can obtain the most information in **debug** logs and the least information in **error** logs.

The following describes each log level in detail.

- **debug**: Debugging level. If this level is set, all log information will be printed.

- **info:** Information level. If this level is set, information about logs of the **warn** level and time consumed for each HTTP/HTTPS request will be printed.
- **warn:** Warning level. If this level is set, information about logs of the error level and information about partial critical events will be printed.
- **error:** Error level. If this level is set, only error information will be printed.

18.4 Missing Modules (SDK for Node.js)

Problem: When OBS Node.js SDK is used for secondary development, an error message was displayed indicating that a module was missing, for example, "Cannot find module 'xml2js'".

Solution:

1. Check whether the version of the local Node.js SDK is the latest.
2. Ensure that the dependency libraries have been correctly installed. For details, see [Installing the SDK \(SDK for Node.js\)](#).

18.5 Connection Timeout (SDK for Node.js)

Problem: If a "connect ETIMEDOUT" error occurred when calling an API, the possible cause was that the service address (endpoint) was incorrect or the network was disconnected.

Solution:

1. Check whether the service address was left blank or invalid.
2. Check the network connection.
3. Check whether the server had returned any exceptions and if so, contact the server-side O&M team.

18.6 Unmatched Signatures (SDK for Node.js)

Problem: The HTTP status code obtained from **CommonMsg.Status** was **403**, and the OBS server-side error code obtained from **CommonMsg.Code** was **SignatureDoesNotMatch**.

Solution:

1. Check whether a bucket name was added before the endpoint. This will lead to a signature mismatch.
2. Ensure that the AK and SK matched, were both correctly specified, and were the same as those used in the request.
3. Check StringToSign.

The structure of StringToSign should be as follows:

```
HTTP-Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedHeaders + CanonicalizedResource
```

19 FAQs (SDK for Node.js)

19.1 How Do I Specify Content-SHA256? (SDK for Node.js)

The **x-obs-content-sha256** header can be carried during object or part upload. Its value is a hexadecimal representation of the SHA-256 value of the request body calculated using **Hex(SHA256Hash(<payload>))**. The server verifies the calculated value for an integrity check. This might affect the performance but is still recommended for security purposes. The sample code for uploading an object is as follows:

```
// Import the OBS library.
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
var crypto = require('crypto');
var fs = require('fs');
// Use the source code to install the client.
// var ObsClient = require('./lib/obs');

// Create an ObsClient instance.
const obsClient = new ObsClient({
  //Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard
  coding may result in leakage.
  //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.ACCESS_KEY_ID,
  secret_access_key: process.env.SECRET_ACCESS_KEY,
  //CN-Hong Kong is used here as an example. Replace it with the one in your actual situation.
  server: 'https://obs.ap-southeast-1.myhuaweicloud.com'
});

let filePath = 'D:\\example.xml'
let sha256 = crypto.createHash('sha256').update(fs.readFileSync(filePath, 'utf8'), 'utf8').digest('hex')

obsClient.putObject({
  Bucket : 'bucketname',
  Key : 'objectname',
  SourceFile : filePath, // filePath indicates the path of the local file to be uploaded, which must have
  the file name included.
  ContentSha256 : sha256
}, (err, result) => {
  if(err){
    console.error('Error-->' + err);
  }else{
  }
```

```
        console.log('Status-->' + result.CommonMsg.Status);
    }
});
```

NOTICE

SDK for Node.js supports both MD5 and SHA256 verification, but SHA256 is recommended for security reasons.

19.2 Why Does the SDK Source Code Contain `acs.amazonaws.com`? (SDK for Node.js)

To be compatible with AWS S3, the source code of OBS SDK for Go contains the **acs.amazonaws.com** keyword. The SDK only takes this keyword as a constant and does not process or access it. You can ignore this keyword. The figure below gives an example. For details, see [SDK for Node.js](#).

```
Utils.prototype.URIAdapter = function(value, signatureContext){
    value = value || '';
    value = String(value);
    if(signatureContext.signature === 'obs'){
        if(obsAllowedUri.indexOf(value) >= 0){
            return value;
        }
        if(value === 'AllUsers' || value === 'http://acs.amazonaws.com/groups/global/AllUsers'){
            return 'Everyone';
        }
    }

    return '';
}

if(v2AllowedUri.indexOf(value) >= 0){
    return value;
}

if(value === 'Everyone' || value === 'AllUsers'){
    return 'http://acs.amazonaws.com/groups/global/AllUsers';
}

if(value === 'AuthenticatedUsers'){
    return 'http://acs.amazonaws.com/groups/global/AuthenticatedUsers';
}

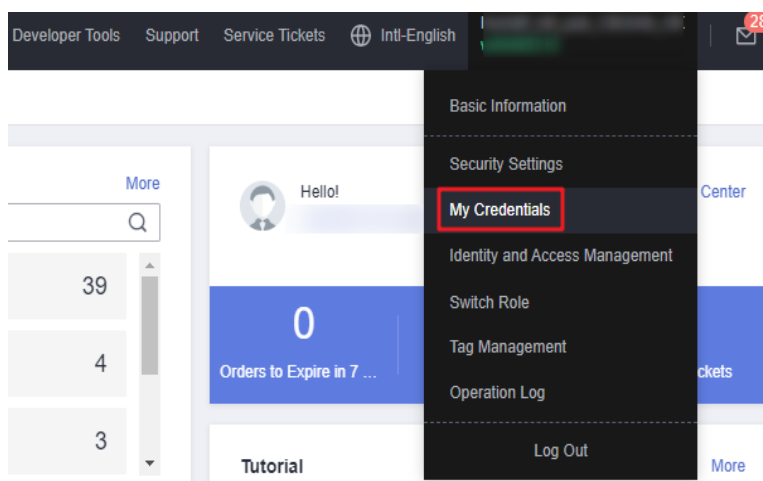
if(value === 'LogDelivery'){
    return 'http://acs.amazonaws.com/groups/s3/LogDelivery';
}

return '';
```

19.3 How Do I Get My Account ID and User ID? (SDK for Node.js)

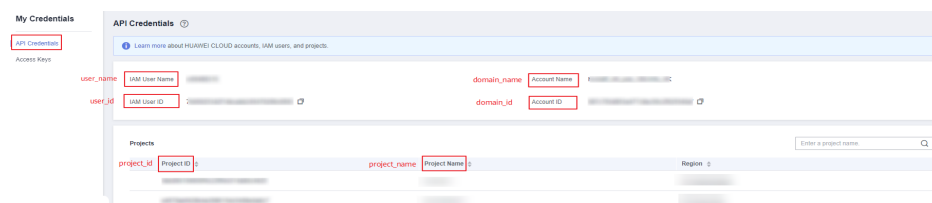
Obtaining Account, IAM User, and Project Information

- Using the console
 - a. On the Huawei Cloud homepage, click **Console** in the upper right corner.
 - b. In the upper right corner, hover over the username and choose **My Credentials** from the drop-down list.

Figure 19-1 My Credentials

- c. On the **API Credentials** page, view the account name, account ID, IAM user name, IAM user ID, project name, and project ID.

The project ID varies depending on the region where your service is located.

Figure 19-2 Viewing the account, user, and project information

- **Calling an API**
 - To obtain a user ID, see [Listing IAM Users](#).
 - To obtain a project ID, see [Querying Project Information](#).

Obtaining User Group Information

Step 1 Log in to the Huawei Cloud console, access the IAM console, and choose **User Groups** in the navigation pane.

Step 2 Expand the details of the desired user group and view its name and ID.

----End

Obtaining Region Information

Step 1 Log in to the Huawei Cloud console, access the IAM console, and choose **Projects** in the navigation pane.

Step 2 View the content in the **Project Name** column. The content in this column indicates the ID of the region that the project belongs to.

----End

Obtaining Agency Information

- Step 1** Log in to the Huawei Cloud console, access the IAM console, and choose **Agencies** in the navigation pane.
- Step 2** Hover over the desired agency to view its name (in the first line) and ID (in the second line) in the dark pop-up box.

----End

19.4 What Is Content-Type (MIME)? (SDK for Node.js)

Multipurpose Internet Mail Extensions (MIME) type is a standard way of describing a data type. The MIME type is passed in the Content-Type header.

NOTE

If you do not specify Content-Type when uploading an object, the SDK determines the object type based on the suffix of the specified object name and automatically assigns a value to Content-Type.

Content-Type

Content-Type indicates the type of data to send or receive and determine the form and encoding method browsers will use to display data (mainly custom client files or media files). If no content type is specified, the type will be generated based on the file name extension. If there is no extension, the content type is **application/octet-stream** by default.

Common Content Types

The commonly used content types are given in the table below.

Table 19-1 List of common content types

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.* (binary stream, unknown file type)	application/octet-stream	.tif	image/tiff
.a11	application/x-a11	.acp	audio/x-mei-aac
.ai	application/postscript	.aif	audio/aiff
.aifc	audio/aiff	.aiff	audio/aiff
.anv	application/x-anv	.apk	application/vnd.android.package-archive
.asa	text/asa	.asf	video/x-ms-asf

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/vnd.adobe.workflow	.biz	text/xml
.bmp	application/x-bmp	.bot	application/x-bot
.c4t	application/x-c4t	.c90	application/x-c90
.cal	application/x-cals	.cat	application/vnd.ms-pki.seccat
.cdf	application/x-netcdf	.cdr	application/x-cdr
.cel	application/x-cel	.cer	application/x-x509-ca-cert
.cg4	application/x-g4	.cgm	application/x-cgm
.cit	application/x-cit	.class	java/
.cml	text/xml	.cmp	application/x-cmp
.cmx	application/x-cmx	.cot	application/x-cot
.crl	application/pkix-crl	.crt	application/x-x509-ca-cert
.csi	application/x-csi	.css	text/css
.csv	text/csv	.cut	application/x-cut
.dbf	application/x-dbf	.dbm	application/x-dbm
.dbx	application/x-dbx	.dcd	text/xml
.dcx	application/x-dcx	.der	application/x-x509-ca-cert
.dgn	application/x-dgn	.dib	application/x-dib
.dll	application/x-msdownload	.doc	application/msword
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	.dot	application/msword

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template	.drw	application/x-drw
.dtd	text/xml	.dwf	Model/vnd.dwf
.dwf	application/x-dwf	.dwg	application/x-dwg
.dxb	application/x-dxb	.dxf	application/x-dxf
.edn	application/vnd.adobe.edn	.emf	application/x-emf
.eml	message/rfc822	.ent	text/xml
.epi	application/x-epi	.eps	application/x-ps
.eps	application/postscript	.etd	application/x-ebx
.exe	application/x-msdownload	.fax	image/fax
.fdf	application/vnd.fdf	.fif	application/fractals
.fo	text/xml	.frm	application/x-frm
.g4	application/x-g4	.gbr	application/x-gbr
.	application/x-	.gif	image/gif
.gl2	application/x-gl2	.gp4	application/x-gp4
.hgl	application/x-hgl	.hmr	application/x-hmr
.hpg	application/x-hpgl	.hpl	application/x-hpl
.hqx	application/mac-binhex40	.hrf	application/x-hrf
.hta	application/hta	.htc	text/x-component
.htm	text/html	.html	text/html
.htt	text/webviewhtml	.htx	text/html
.icb	application/x-icb	.ico	image/x-icon
.ico	application/x-ico	.iff	application/x-iff
.ig4	application/x-g4	.igs	application/x-igs

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.iii	application/x-iphone	.img	application/x-img
.ins	application/x-internet-signup	.ipa	application/vnd.iphone
.isp	application/x-internet-signup	.IVF	video/x-ivf
.java	java/*	.jfif	image/jpeg
.jpe	image/jpeg	.jpe	application/x-jpe
.jpeg	image/jpeg	.jpg	image/jpeg
.jpg	application/x-jpg	.js	application/x-javascript
.jsp	text/html	.la1	audio/x-liquid-file
.lar	application/x-laplayer-reg	.latex	application/x-latex
.lavs	audio/x-liquid-secure	.lbm	application/x-lbm
.lmsff	audio/x-la-lms	.ls	application/x-javascript
.ltr	application/x-ltr	.m1v	video/x-mpeg
.m2v	video/x-mpeg	.m3u	audio/mpegurl
.m4e	video/mpeg4	.mac	application/x-mac
.man	application/x-troff-man	.math	text/xml
.mdb	application/msaccess	.mdb	application/x-mdb
.mfp	application/x-shockwave-flash	.mht	message/rfc822
.mhtml	message/rfc822	.mi	application/x-mi
.mid	audio/mid	.midi	audio/mid
.mil	application/x-mil	.mml	text/xml
.mnd	audio/x-musicnet-download	.mns	audio/x-musicnet-stream
.mocha	application/x-javascript	.mov	video/quicktime

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.movie	video/x-sgi-movie	mp1	audio/mp1
.mp2	audio/mp2	.mp2v	video/mpeg
.mp3	audio/mp3	.mp4	video/mp4
.mpa	video/x-mpg	.mpd	application/vnd.ms-project
.mpe	video/x-mpeg	.mpeg	video/mpg
.mpg	video/mpg	.mpga	audio/rn-mpeg
.mpp	application/vnd.ms-project	.mps	video/x-mpeg
.mpt	application/vnd.ms-project	.mpv	video/mpg
.mpv2	video/mpeg	.mpw	application/vnd.ms-project
.mpx	application/vnd.ms-project	.mtx	text/xml
.mxp	application/x-mmxp	.net	image/pnetvue
.nrf	application/x-nrf	.nws	message/rfc822
.odc	text/x-ms-odc	.out	application/x-out
.p10	application/pkcs10	.p12	application/x-pkcs12
.p7b	application/x-pkcs7-certificates	.p7c	application/pkcs7-mime
.p7m	application/pkcs7-mime	.p7r	application/x-pkcs7-certreqresp
.p7s	application/pkcs7-signature	.pc5	application/x-pc5
.pci	application/x-pci	.pcl	application/x-pcl
.pcx	application/x-pcx	.pdf	application/pdf
.pdb	chemical/x-pdb	.pdx	application/vnd.adobe.pdx
.pfx	application/x-pkcs12	.pgl	application/x-pgl

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.pic	application/x-pic	.pko	application/vnd.ms-pki.pko
.pl	application/x-perl	.plg	text/html
.pls	audio/scpls	.plt	application/x-plt
.png	image/png	.png	application/x-png
.pot	application/vnd.ms-powerpoint	.potx	application/vnd.openxmlformats-officedocument.presentationml.template
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppsx	application/vnd.openxmlformats-officedocument.presentationml.slide-show
.ppt	application/vnd.ms-powerpoint	.ppt	application/x-ppt
.pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation	.pr	application/x-pr
.prf	application/pics-rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/postscript	.ptn	application/x-ptn
.pwz	application/vnd.ms-powerpoint	.r3t	text/vnd.rn-realtex3d
.ra	audio/vnd.rn-realaudio	.ram	audio/x-pn-realaudio

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.ras	application/x-ras	.rat	application/rat-file
.rdf	text/xml	.rec	application/vnd.rn-recording
.red	application/x-red	.rgb	application/x-rgb
.rjs	application/vnd.rn-realsystem-rjs	.rjt	application/vnd.rn-realsystem-rjt
.rlc	application/x-rlc	.rle	application/x-rle
.rm	application/vnd.rn-realmedia	.rmf	application/vnd.adobe.rmf
.rmi	audio/mid	.rmj	application/vnd.rn-realsystem-rmj
.rmm	audio/x-pn-realaudio	.rmp	application/vnd.rn-rn_music_package
.rms	application/vnd.rn-realmedia-secure	.rmvb	application/vnd.rn-realmedia-vbr
.rmx	application/vnd.rn-realsystem-rmx	.rnx	application/vnd.rn-realplayer
.rp	image/vnd.rn-realpix	.rpm	audio/x-pn-realaudio-plugin
.rsml	application/vnd.rn-rsml	.rt	text/vnd.rn-realtex
.rtf	application/msword	.rtf	application/x-rtf
.rv	video/vnd.rn-realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp
.sdw	application/x-sdw	.sis	application/vnd.symbian.install

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.sisx	application/vnd.symbian.install	.sit	application/x-stuffit
.slb	application/x-slb	.sld	application/x-sld
.sldx	application/vnd.openxmlformats-officedocument.presentationml.slide	.slk	drawing/x-slk
.smi	application/smil	.smil	application/smil
.smk	application/x-smk	.snd	audio/basic
.sol	text/plain	.sor	text/plain
.spc	application/x-pkcs7-certificates	.spl	application/futuresplash
.spp	text/xml	.ssm	application/streamingmedia
.sst	application/vnd.ms-pki.certstore	.stl	application/vnd.ms-pki.stl
.stm	text/html	.sty	application/x-sty
.svg	image/svg+xml	.swf	application/x-shockwave-flash
.tdf	application/x-tdf	.tg4	application/x-tg4
.tga	application/x-tga	.tif	image/tiff
.tif	application/x-tif	.tiff	image/tiff
.tld	text/xml	.top	drawing/x-top
.torrent	application/x-bittorrent	.tsd	text/xml
.txt	text/plain	.uin	application/x-icq
.uls	text/iuls	.vcf	text/x-vcard
.vda	application/x-vda	.vdx	application/vnd.visio
.vml	text/xml	.vpg	application/x-mpeg005

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.vsd	application/vnd.visio	.vsd	application/x-vsd
.vss	application/vnd.visio	.vst	application/vnd.visio
.vst	application/x-vst	.vsw	application/vnd.visio
.vsx	application/vnd.visio	.vtx	application/vnd.visio
.vxml	text/xml	.wav	audio/wav
.wax	audio/x-ms-wax	.wb1	application/x-wb1
.wb2	application/x-wb2	.wb3	application/x-wb3
.wbmp	image/vnd.wap.wbmp	.wiz	application/msword
.wk3	application/x-wk3	.wk4	application/x-wk4
.wkq	application/x-wkq	.wks	application/x-wks
.wm	video/x-ms-wm	.wma	audio/x-ms-wma
.wmd	application/x-ms-wmd	.wmf	application/x-wmf
.wml	text/vnd.wap.wml	.wmv	video/x-ms-wmv
.wmx	video/x-ms-wmx	.wmz	application/x-ms-wmz
.wp6	application/x-wp6	.wpd	application/x-wpd
.wpg	application/x-wpg	.wpl	application/vnd.ms-wpl
.wq1	application/x-wq1	.wr1	application/x-wr1
.wri	application/x-wri	.wrk	application/x-wrk
.ws	application/x-ws	.ws2	application/x-ws
.wsc	text/scriptlet	.wsdl	text/xml
.wvx	video/x-ms-wvx	.xap	application/x-silverlight-app
.x_b	application/x-x_b	.xdp	application/vnd.adobe.xdp

File Name Extension	Content-Type(Mime-Type)	File Name Extension	Content-Type(Mime-Type)
.xdr	text/xml	.xfd	application/vnd.adobe.xfd
.xdf	application/vnd.adobe.xdf	.xhtml	text/html
.xls	application/vnd.ms-excel	.xls	application/x-xls
.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	.xltx	application/vnd.openxmlformats-officedocument.spreadsheetml.template
.xlw	application/x-xlw	.xml	text/xml
.xpl	audio/scpls	.xq	text/xml
.xql	text/xml	.xquery	text/xml
.xsd	text/xml	.xsl	text/xml
.xslt	text/xml	.xwd	application/x-xwd
.x_t	application/x-x_t	.yaml	text/vnd.yaml
.yaml	text/vnd.yaml	.webp	image/webp
.tar	application/x-tar	.zip	application/zip