

Object Storage Service

Python SDK Developer Guide (Ally Region)

Issue	01
Date	2026-01-15



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. 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.

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 Before You Start (SDK for Python)	1
2 API Overview of OBS SDK for Python	2
3 Preparations (SDK for Python)	7
4 Downloading and Installing an SDK (SDK for Python)	9
5 Getting Started (SDK for Python)	11
6 Initialization (SDK for Python)	15
6.1 Initializing an Instance of ObsClient (SDK for Python)	15
6.2 Initializing a bucketClient Instance (SDK for Python)	21
6.3 Log Initialization (SDK for Python)	22
6.4 SDK Common Result Objects (SDK for Python)	24
7 Bucket-Related APIs (SDK for Python)	27
7.1 Creating a Bucket (SDK for Python)	27
7.2 Obtaining a Bucket List (SDK for Python)	37
7.3 Checking Whether a Bucket Exists (SDK for Python)	42
7.4 Deleting a Bucket (SDK for Python)	46
7.5 Listing Objects in a Bucket (SDK for Python)	50
7.6 Listing Object Versions in a Bucket (SDK for Python)	67
7.7 Obtaining Bucket Metadata (SDK for Python)	85
7.8 Obtaining the Region of a Bucket (SDK for Python)	94
7.9 Obtaining Storage Information of a Bucket (SDK for Python)	99
7.10 Configuring a Storage Quota (SDK for Python)	104
7.11 Obtaining a Bucket Storage Quota (SDK for Python)	109
7.12 Configuring a Storage Class for a Bucket (SDK for Python)	114
7.13 Obtaining the Storage Class of a Bucket (SDK for Python)	119
7.14 Configuring a Bucket ACL (SDK for Python)	124
7.15 Obtaining a Bucket ACL (SDK for Python)	135
7.16 Configuring Logging for a Bucket (SDK for Python)	144
7.17 Obtaining the Logging Configuration of a Bucket (SDK for Python)	153
7.18 Configuring a Bucket Policy (SDK for Python)	162
7.19 Obtaining the Policy of a Bucket (SDK for Python)	167
7.20 Deleting a Bucket Policy (SDK for Python)	172

7.21 Configuring Lifecycle Rules for a Bucket (SDK for Python).....	176
7.22 Obtaining the Lifecycle Configuration of a Bucket (SDK for Python).....	192
7.23 Deleting the Lifecycle Configuration of a Bucket (SDK for Python).....	206
7.24 Configuring Static Website Hosting for a Bucket (SDK for Python).....	210
7.25 Obtaining Static Website Hosting for a Bucket (SDK for Python).....	220
7.26 Deleting Static Website Hosting for a Bucket (SDK for Python).....	229
7.27 Configuring Versioning for a Bucket (SDK for Python).....	234
7.28 Obtaining the Versioning Status of a Bucket (SDK for Python).....	239
7.29 Configuring CORS for a Bucket (SDK for Python).....	243
7.30 Obtaining the CORS Configuration of a Bucket (SDK for Python).....	251
7.31 Deleting the CORS Configuration of a Bucket (SDK for Python).....	257
7.32 Configuring Tags for a Bucket (SDK for Python).....	262
7.33 Obtaining Bucket Tags (SDK for Python).....	268
7.34 Deleting Bucket Tags (SDK for Python).....	274
7.35 Obtaining the Custom Domain Name of a Bucket (SDK for Python).....	278
8 Object-Related APIs (SDK for Python).....	286
8.1 Object Upload Overview (SDK for Python).....	286
8.2 Uploading Objects - Text-Based (SDK for Python).....	287
8.3 Uploading an Object - Streaming (SDK for Python).....	304
8.4 Uploading an Object - File-Based (SDK for Python).....	321
8.5 Uploading an Object - Append (SDK for Python).....	339
8.6 Uploading an Object - Resumable (SDK for Python).....	360
8.7 Obtaining the Upload Progress (SDK for Python).....	377
8.8 Uploading an Object - Browser-Based (SDK for Python).....	378
8.9 Object Download Overview (SDK for Python).....	383
8.10 Downloading an Object - Binary (SDK for Python).....	383
8.11 Downloading an Object - Streaming (SDK for Python).....	402
8.12 Downloading an Object - File-Based (SDK for Python).....	421
8.13 Downloading an Object - Range-Based (SDK for Python).....	440
8.14 Downloading an Object - Resumable (SDK for Python).....	458
8.15 Downloading an Object - Obtaining the Download Progress (SDK for Python).....	475
8.16 Uploading an Object - Creating a Folder (SDK for Python).....	476
8.17 Copying an Object (SDK for Python).....	492
8.18 Deleting an Object (SDK for Python).....	510
8.19 Batch Deleting Objects (SDK for Python).....	516
8.20 Obtaining Object Metadata (SDK for Python).....	524
8.21 Modifying Object Metadata (SDK for Python).....	536
8.22 Configuring an Object ACL (SDK for Python).....	546
8.23 Obtaining an Object ACL (SDK for Python).....	558
8.24 Restoring a Cold Object (SDK for Python).....	568
8.25 Using CRC64 to Verify File Consistency (SDK for Python).....	574
9 APIs Related to Multipart Upload (SDK for Python).....	584

9.1 Multipart Upload Overview (SDK for Python).....	584
9.2 Initiating a Multipart Upload (SDK for Python).....	586
9.3 Uploading a Part (SDK for Python).....	600
9.4 Listing Uploaded Parts (SDK for Python).....	609
9.5 Listing Multipart Uploads (SDK for Python).....	621
9.6 Assembling Parts (SDK for Python).....	636
9.7 Copying a Part (SDK for Python).....	645
9.8 Aborting a Multipart Upload (SDK for Python).....	653
10 Client-Side Encryption APIs (SDK for Python).....	659
10.1 Overview (SDK for Python).....	659
10.2 Client-Side Encryption APIs (SDK for Python).....	661
11 Other APIs (SDK for Python).....	672
11.1 Creating a Signed URL (SDK for Python).....	672
11.2 Generating Authentication Parameters for Browser-Based Uploads (SDK for Python).....	678
11.3 Server-Side Encryption (SDK for Python).....	682
11.4 Static Website Hosting (SDK for Python).....	688
11.5 User-defined Headers (SDK for Python).....	692
12 Troubleshooting (SDK for Python).....	696
12.1 HTTP Status Codes (SDK for Python).....	696
12.2 OBS Server-Side Error Codes (SDK for Python).....	698
12.3 Log Analysis (SDK for Python).....	707
13 FAQs (SDK for Python).....	709
13.1 How Do I Make an Object Accessible to Anonymous Users? (SDK for Python).....	709
13.2 How Do I Obtain an Object URL? (Python SDK).....	709
13.3 How Do I Improve the Uploading Speed of Large Files over the Public Network? (Python SDK).....	710
13.4 Setting an Object Expiration Time (SDK for Python).....	710
13.5 What Is Content-Type (MIME)? (Python SDK).....	711
13.6 How Do I Get My Account ID and IAM User ID? (SDK for Python).....	720

1 Before You Start (SDK for Python)

This section describes the important notes about Object Storage Service (OBS) SDK for Python.

Important Notes

- Some features are available only in some regions. If **405** HTTP status code is returned for a call of a certain feature API, check whether the region supports this feature.
- The namespace is incompatible with earlier OBS versions (2.1.x). All modules are saved in the **obs** package.
- API functions are compatible with earlier OBS versions (2.1.x).

2 API Overview of OBS SDK for Python

[Table 2-1](#) describes the APIs provided by OBS SDK for Python. You can click an API name in the table to see its detailed information and sample code.

SDK API Overview

Table 2-1 APIs provided by OBS SDK for Python

Interface	Method	Function
Creating a Bucket (SDK for Python)	ObsClient.createBucket	Creates a bucket.
Obtaining a Bucket List (SDK for Python)	ObsClient.listBuckets	Returns a list of buckets in alphabetical order.
Checking Whether a Bucket Exists (SDK for Python)	ObsClient.headBucket	Determines whether a bucket exists.
Deleting a Bucket (SDK for Python)	ObsClient.deleteBucket	Deletes an empty bucket.
Listing Objects in a Bucket (SDK for Python)	ObsClient.listObjects	Returns some or all (up to 1,000) of the objects in a bucket.
Listing Object Versions in a Bucket (SDK for Python)	ObsClient.listVersions	Lists some or all (up to 1,000) of the object versions in a bucket.
Listing Multipart Uploads (SDK for Python)	ObsClient.listMultipartUploads	Lists ongoing multipart uploads.

Interface	Method	Function
Obtaining Bucket Metadata (SDK for Python)	ObsClient.getBucketMeta data	Sends a HEAD request to a bucket to obtain the bucket metadata such as CORS rules (if set), the storage class, and other information.
Obtaining the Region of a Bucket (SDK for Python)	ObsClient.getBucketLoca tion	Returns the region where the bucket is created.
Obtaining Storage Information of a Bucket (SDK for Python)	ObsClient.getBucketStora geInfo	Returns the storage information of a bucket, including the number of objects and the space occupied by the objects in the bucket.
Configuring a Storage Quota (SDK for Python)	ObsClient.setBucketQuot a	Sets a storage quota for a bucket.
Obtaining a Bucket Storage Quota (SDK for Python)	ObsClient.getBucketQuot a	Returns the storage quota of a bucket.
Configuring a Storage Class for a Bucket (SDK for Python)	ObsClient.setBucketStora gePolicy	Configures a storage class for a bucket.
Obtaining the Storage Class of a Bucket (SDK for Python)	ObsClient.getBucketStora gePolicy	Returns the storage class of a bucket.
Configuring a Bucket ACL (SDK for Python)	ObsClient.setBucketAcl	Configures a bucket ACL.
Obtaining a Bucket ACL (SDK for Python)	ObsClient.getBucketAcl	Obtains a bucket ACL.
Configuring Logging for a Bucket (SDK for Python)	ObsClient.setBucketLoggi ng	Configures logging for a bucket.
Obtaining the Logging Configuration of a Bucket (SDK for Python)	ObsClient.getBucketLogg ing	Returns the logging configuration of a bucket.
Configuring a Bucket Policy (SDK for Python)	ObsClient.setBucketPolic y	Configures a bucket policy. If the bucket already has a policy, the policy will be overwritten by the one specified in this request.

Interface	Method	Function
Obtaining the Policy of a Bucket (SDK for Python)	ObsClient.getBucketPolicy	Returns the policy of a bucket.
Deleting a Bucket Policy (SDK for Python)	ObsClient.deleteBucketPolicy	Deletes the policy of a bucket.
Configuring Lifecycle Rules for a Bucket (SDK for Python)	ObsClient.setBucketLifecycle	Configures lifecycle rules for a bucket to periodically transition objects between storage classes or delete objects.
Obtaining the Lifecycle Configuration of a Bucket (SDK for Python)	ObsClient.getBucketLifecycle	Returns the lifecycle configuration of a bucket.
Deleting the Lifecycle Configuration of a Bucket (SDK for Python)	ObsClient.deleteBucketLifecycle	Deletes all lifecycle rules of a bucket.
Configuring Static Website Hosting for a Bucket (SDK for Python)	ObsClient.setBucketWebsite	Configures static website hosting for a bucket.
Obtaining Static Website Hosting for a Bucket (SDK for Python)	ObsClient.getBucketWebsite	Returns the static website hosting configuration of a bucket.
Deleting Static Website Hosting for a Bucket (SDK for Python)	ObsClient.deleteBucketWebsite	Deletes the static website hosting configurations of a bucket.
Configuring Versioning for a Bucket (SDK for Python)	ObsClient.setBucketVersioning	Configures versioning for a bucket.
Obtaining the Versioning Status of a Bucket (SDK for Python)	ObsClient.getBucketVersioning	Returns the versioning status of a bucket.
Configuring CORS for a Bucket (SDK for Python)	ObsClient.setBucketCors	Configures CORS for a bucket to allow cross-origin requests.

Interface	Method	Function
Obtaining the CORS Configuration of a Bucket (SDK for Python)	ObsClient.getBucketCors	Returns the CORS rules of a bucket.
Deleting the CORS Configuration of a Bucket (SDK for Python)	ObsClient.deleteBucketCors	Deletes the CORS configuration of a bucket.
Object Upload Overview (SDK for Python)	ObsClient.putContent	Uploads an object to a bucket.
Uploading an Object - File-Based (SDK for Python)	ObsClient.putFile	Uploads a file or folder to a bucket.
Uploading an Object - Append (SDK for Python)	ObsClient.appendObject	Appends content to an uploaded object.
Object Download Overview (SDK for Python)	ObsClient.getObject	Downloads an object from a bucket.
Copying an Object (SDK for Python)	ObsClient.copyObject	Creates a copy for a specified object.
Deleting an Object (SDK for Python)	ObsClient.deleteObject	Deletes a single object from a bucket.
Batch Deleting Objects (SDK for Python)	ObsClient.deleteObjects	Deletes multiple objects from a bucket in a batch.
Obtaining Object Metadata (SDK for Python)	ObsClient.getObjectMetadata	Returns the object metadata.
Modifying Object Metadata (SDK for Python)	ObsClient.setObjectMetadata	Modifies the metadata for an object.
Configuring an Object ACL (SDK for Python)	ObsClient.setObjectAcl	Configures an ACL for an object in a specified bucket.
Obtaining an Object ACL (SDK for Python)	ObsClient.getObjectAcl	Obtains the ACL of an object in a specified bucket.
Initiating a Multipart Upload (SDK for Python)	ObsClient.initiateMultipartUpload	Initiates a multipart upload in a bucket.

Interface	Method	Function
Uploading a Part (SDK for Python)	ObsClient.uploadPart	Uploads parts to a bucket by specifying the multipart upload ID.
Copying a Part (SDK for Python)	ObsClient.copyPart	Copies a part to a specified bucket by specifying the multipart upload ID.
Listing Uploaded Parts (SDK for Python)	ObsClient.listParts	Returns the uploaded parts in a bucket by specifying the multipart upload ID.
Assembling Parts (SDK for Python)	ObsClient.completeMultipartUpload	Completes a multipart upload based on the multipart upload ID.
Aborting a Multipart Upload (SDK for Python)	ObsClient.abortMultipartUpload	Aborts a multipart upload in a bucket by specifying the multipart upload ID.
Restoring a Cold Object	ObsClient.restoreObject	Restores a Cold object.
Creating a Signed URL (SDK for Python)	ObsClient.createSignedUrl	Creates a signed URL based on the specified AK and SK, HTTP method, and request parameters.
Generating Authentication Parameters for Browser-Based Uploads (SDK for Python)	ObsClient.createPostSignature	Creates request parameters for authentication to enable browser-based upload with POST.
Uploading an Object - Resumable (SDK for Python)	ObsClient.uploadFile	Enables resumable uploads of objects in the event of unstable network connections or program crashes.
Downloading an Object - Resumable (SDK for Python)	ObsClient.downloadFile	Adds the resumable function to the partial download API.

3 Preparations (SDK for Python)

Before using OBS SDK for Python to access OBS, you need to prepare the service and development environments. To prepare the service environment, you must get an account and an access key. Both of them are necessary for interaction between OBS SDK for Python and OBS. To ensure successful SDK installation and SDK-based code development and running, you should also set up a local development environment, for example, installing dependencies and development tools.

Preparing Access Keys

Access keys consist of two parts: an access key ID (AK) and a secret access key (SK). OBS uses access keys to sign requests to make sure that only authorized accounts can access specified OBS resources. Programmatic access must be enabled for an IAM user before the IAM user can get access keys. Access keys are explained as follows:

- One AK maps to only one user but one user can have multiple AKs. OBS authenticates users by their AKs.
- An SK is required for accessing OBS. Authentication information is generated based on the SK and request headers. AKs and SKs are in one-to-one match.

Access keys are classified into permanent access keys (AK/SK) and temporary access keys (AK/SK and security token). Each user can create at most two permanent access keys. Temporary access keys must be used within a given validity period. Once expired, they must be requested again. For security purposes, you are advised to use temporary access keys to access OBS. If you want to use permanent access keys, periodically update them. The following describes how to obtain two types of access keys.

- To get permanent access keys, do as follows:
 - a. Log in to the management console.
 - b. In the upper right corner, hover 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**.
 - e. In the displayed dialog box, enter the login password and verification code.

 NOTE

- If you have not bound an email address or a mobile number yet, only the login password is required.
 - If you have bound both an email address and a mobile number, you can use either of them for verification.
- f. Click **OK**.
 - g. Click **Download**. The access key file is automatically saved to your browser's default download path.
 - h. Open the downloaded **credentials.csv** file to obtain the access keys (AK and SK).

 NOTE

- Each user can create a maximum of two valid access key pairs.
 - Keep AKs and SKs 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 a new AK/SK pair if needed.
- To get temporary access keys, refer to the following:
Temporary access keys are issued by the system and are only valid for 15 minutes to 24 hours. Once expired, they must be requested again. They follow the principle of least privilege. When a temporary AK/SK pair is used for authentication, a security token must be used at the same time.

Setting Up a Development Environment

- Download a proper Python version from the [Python official website](#) and install it.
 - Recommended Python 2.x version: 2.7.x
 - Recommended Python 3.x versions: 3.6, 3.7, 3.8, 3.9, 3.10, and 3.11

 NOTE

Python 3.5 and earlier versions are not recommended. If you need to use these versions, run the **pip install secrets** command to install the secrets module. You can install python2-secrets on Python 2.7.

- Download the latest community version of PyCharm from the [PyCharm official website](#).
- Run **pip install pycryptodome==3.10.1** to install the cryptographic library.

4 Downloading and Installing an SDK (SDK for Python)

This topic provides the download links and installation methods of OBS SDK for Python.

Downloading OBS SDK for Python

Latest version of OBS Python SDK source code: [Download](#)

Installing OBS SDK for Python

You can use the methods listed in [Table 4-1](#) to install OBS SDK for Python.

Table 4-1 Methods of installing OBS SDK for Python

No.	Method
2	Installing the SDK Using the Source Code
3	Installing the SDK Using setuptools

Method: Installing the SDK Using the Source Code

The following procedures show an example of installing the latest version of OBS Python SDK.

- Step 1** the SDK package and decompress it.
- Step 2** Run `pip install pycryptodome==3.10.1` to install the cryptographic library.
- Step 3** Decompress the development package to obtain folder **src** (SDK source code), folder **examples** (sample code), file **README.md** (feature description file of SDK versions), and file **log.conf** (SDK log configuration file).
- Step 4** Use PyCharm to create a project, copy the folders and files obtained in the previous step to the project, right-click folder **src**, and choose **Mark Directory as > Sources Root**.

 NOTE

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

```
|— examples
|— src
|— log.conf
|— README.md
```

----End

Method: Installing the SDK Using setuptools

The following procedures show an example of installing the latest version of OBS Python SDK.

- Step 1** the SDK package and decompress it.
- Step 2** Download and install [setuptools](#).
- Step 3** On the command-line interface (CLI), go to folder **src** under the directory where the development package is decompressed.
- Step 4** Run the **python setup.py install** command to install the SDK.
- Step 5** After the installation, check whether a folder named **esdk_obs_python-<versionId>-***.egg**** is generated in **Lib/site-package** under the Python installation directory.

 NOTE

- If you use this method to install the SDK, you need to delete folder **esdk_obs_python-<versionId>-***.egg**** when you re-install the SDK.
- If SDK modules cannot be loaded after you have performed the previous steps, you can directly add the absolute path of the **src** directory in OBS Python SDK to the **sys.path** list.

----End

5 Getting Started (SDK for Python)

Creating an AK and SK

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**.
 - e. In the **Create Access Key** dialog box, enter a description (recommended), and click **OK**.
 - f. In the displayed dialog box, click **Download** to save the access keys to your browser's default download path.
 - g. 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.

Initializing an Instance of ObsClient

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

```
# Import the module.
from obs import ObsClient

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
# Use the instance to access OBS.

# Close ObsClient.
obsClient.close()
```

NOTE

For more information, see chapter [Initialization](#).

For details about log configuration, see [Log Initialization \(SDK for Python\)](#).

Creating a Bucket

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.

This example creates a bucket named **examplebucket** and specifies its location, ACL, storage class, and redundancy type.

```
from obs import CreateBucketHeader, HeadPermission
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
```

```
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Add additional headers to specify a private bucket in the Standard storage class that supports multi-AZ
    storage.
    header = CreateBucketHeader(aclControl=HeadPermission.PRIVATE, storageClass="STANDARD",
availableZone="3az")
    # Specify the region where the bucket is to be created. The region must be the same as that in the
    endpoint passed.
    location = "region"
    bucketName = "examplebucket"
    # Create a bucket.
    resp = obsClient.createBucket(bucketName, header, location)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Create Bucket Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Create Bucket Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Create Bucket Failed')
    print(traceback.format_exc())
```

NOTE

- Bucket names are globally unique. Ensure that the bucket you create is named differently from any other bucket.
- 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 periods (.) and hyphens (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
- If you repeatedly create buckets of the same name, no error will be reported and the bucket attributes comply with those specified in the first creation request.
- For more information, see [Creating a Bucket](#).

Uploading an Object

This example uploads a text.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
```

```
# Before running the sample code, ensure that the environment variables AccessKeyID and SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the text to be uploaded.
    content = 'Hello OBS'
    # Upload the text.
    resp = obsClient.putContent(bucketName, objectKey, content)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put Content Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
    else:
        print('Put Content Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

NOTE

For more information, see [Object Upload Overview](#).

6 Initialization (SDK for Python)

6.1 Initializing an Instance of ObsClient (SDK for Python)

Function

ObsClient functions as the Python client for accessing OBS. It offers users a series of APIs for interaction with OBS. These APIs are used for managing resources, such as buckets and objects, stored in OBS.

Method

```
ObsClient(access_key_id, secret_access_key, server)
```

Constructor Parameter Description

Parameter	Description	Recommended Value
access_key_id	Access key ID (AK). It is left blank by default, which indicates that anonymous users are allowed for access.	N/A
secret_access_key	Secret access key (SK). It is left blank by default, which indicates that anonymous users are allowed for access.	N/A
security_token	Security token in the temporary access keys.	N/A

Parameter	Description	Recommended Value
server	Server address for accessing OBS. It consists of a protocol type, domain name, 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]
max_redirect_count	Maximum number of times that the HTTP/HTTPS request is redirected. The default value is 10 .	[1,10]
timeout	Timeout period (in seconds) of an HTTP/HTTPS request. The default value is 60 .	[10,60]
ssl_verify	Verification of server certificate parameters to check whether the server certificate matches the client certificate. The options are as follows: - Path to the server-side root certificate file in PEM format True: The certificate list will be obtained from the root certificate library and the certificates of the operating system (Windows only) will be verified. False: The server-side certificates will not be verified. The default value is False .	N/A
custom_ciphers	Algorithm suite used for HTTPS link establishment.	N/A
chunk_size	Chunk size (in bytes) set for reading and writing socket streams. The default value is 65536 .	Default
long_conn_mode	Whether to enable the persistent connection mode. The default value is False .	N/A
proxy_host	Host IP address of the proxy server. This value is left blank by default. NOTE Do not specify http:// or https:// for the proxy server's host address.	N/A

Parameter	Description	Recommended Value
proxy_port	Port number of the proxy server. This value is left blank by default.	N/A
proxy_username	User name used for connecting to the proxy server. This value is left blank by default.	N/A
proxy_password	Password used for connecting to the proxy server. This value is left blank by default.	N/A
is_cname	Whether to use a user-defined domain name to access OBS. The default value is False . NOTE To protect your services from being affected by the takeover of Huawei Cloud public domain names by relevant organizations, you are advised to use a user-defined domain name to access a bucket.	N/A
security_providers	How an access key is obtained. The default value is None . NOTE The value of security_providers must be in a list. The default value None indicates the default search methods to obtain the access keys from the environment variables or from ECSs. If this parameter is specified, the default search methods are not provided. Instead, the search methods specified by security_providers are used.	N/A
security_provider_policy	Specifies the allowed access key search policy. The default value is None . NOTE - This parameter is used to set the search policy. The default value None indicates the specified access keys are displayed. In addition, if the access key parameters are specified, security_provider_policy is ignored. - If security_provider_policy is set to OBS_DEFAULT, the access keys are obtained by searching in sequence. - If security_provider_policy is set to the predefined methods (ENV or ECS), the access keys are obtained using the corresponding method.	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.

Code Examples

- You can create an instance of ObsClient by using a constructor function. Sample code for creating an instance of ObsClient using permanent access keys (AK/SK):

```
# Import the module.
from obs import ObsClient

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways.
Using hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
Specify a security token.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

# Use the instance to access OBS.

# Close ObsClient.
obsClient.close()
```

- Sample code for creating an instance of ObsClient using temporary access keys (AK/SK and security token):

```
# Import the module.
from obs import ObsClient

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways.
Using hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them
from environment variables.
security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(
    access_key_id=ak,
    secret_access_key=sk,
```

```
server=server,
security_token=security_token
)

# Use the instance to access OBS.

# Close ObsClient.
obsClient.close()
```

- Specify the method of obtaining temporary access keys:
 - Sample code for obtaining the access key from environment variables using a single method:

```
# Import the module.
from obs import ObsClient
from obs import loadtoken

# Create an instance of ObsClient.
# Provide ENV to obtain the access keys.
obsClient = ObsClient(
    server='https://your-endpoint',
    security_providers=[loadtoken.ENV]
)

# Use the instance to access OBS.

# Close ObsClient.
obsClient.close()
```

- You can also create an instance of ObsClient by using temporary access keys obtained by configuring system environment variables or by accessing an ECS.

- Sample code for creating an instance of ObsClient using ENV:

```
# Import the module.
from obs import ObsClient

# Create an instance of ObsClient.
# Provide ENV to obtain the access keys.
obsClient = ObsClient(
    server='https://your-endpoint',
    security_provider_policy='ENV'
)

# Use the instance to access OBS.

# Close ObsClient.
obsClient.close()
```

NOTE

In the preceding method, access keys are searched from the environment variables of the current system. The **OBS_ACCESS_KEY_ID** and **OBS_SECRET_ACCESS_KEY** fields need to be defined in the corresponding environment variables. If temporary access keys are used, the **OBS_SECURITY_TOKEN** field must also be defined in the environment variables.

- Sample code for creating an instance of ObsClient using ECS:

```
# Import the module.
from obs import ObsClient

# Create an instance of ObsClient.
# Provide ECS to obtain the temporary access keys.
obsClient = ObsClient(
    server='https://your-endpoint',
    security_provider_policy='ECS'
)

# Use the instance to access OBS.
```



```
# Close ObsClient.  
obsClient.close()
```

NOTE

When an application is deployed on an ECS, temporary access keys can be obtained automatically using the preceding methods and updated periodically.

If the client reports error 401, check whether an agency has been added during ECS creation.

NOTICE

When obtaining temporary access keys using this method, ensure that the UTC time of the server is the same as that of the environment where the application is deployed. Otherwise, the temporary access keys may fail to be updated.

- In addition to the preceding methods, you can also search in sequence to obtain the corresponding access keys from the environment variables and ECSs.
 - You can set **security_provider_policy** to **OBS_DEFAULT** to specify that ObsClient searches for access keys in sequence.

```
# Import the module.  
from obs import ObsClient  
  
# Create an instance of ObsClient.  
# Search for access keys from environment variables and ECSs in sequence.  
obsClient = ObsClient(  
    server='https://your-endpoint',  
    security_provider_policy='OBS_DEFAULT'  
)  
  
# Use the instance to access OBS.  
  
# Close ObsClient.  
obsClient.close()
```

NOTE

In the preceding method, **security_provider_policy** is set to **OBS_DEFAULT**, which specifies that ObsClient searches for access keys in sequence from the predefined list. By default, the system provides two predefined search methods: obtaining the access keys from the environment variables and obtaining from ECSs. ObsClient searches for the access keys from the environment variables first and then from ECSs. In this case, ObsClient is created using the first pair of access keys obtained in the search.

NOTE

- The project can contain one or more instances of **ObsClient**.
- ObsClient is thread secure and can be simultaneously used by multiple threads.

6.2 Initializing a bucketClient Instance (SDK for Python)

Function

BucketClient functions as the Python client for accessing an OBS bucket. It offers users a series of APIs for interaction with OBS. These APIs are used for managing resources, such as buckets and objects, stored in OBS.

Except for **ObsClient.listBuckets**, **ObsClient.downloadFile**, **ObsClient.uploadFile**, **ObsClient.createPostSignature**, and **ObsClient.createSignedUrl**, **BucketClient** implements the same APIs as **ObsClient**, including the same functions and parameters, with the **bucketName** parameter omitted.

Method

```
obsClient.bucketClient(  
    bucketName='*** Your Bucket Name ***'  
)
```

Constructor Parameter Description

Parameter	Type	Mand atory (Yes/N o)	Description
bucket_name	str	Yes	Name of the bucket client to be created

Code Examples

```
# Import the module.  
from obs import ObsClient  
  
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using  
# hard coding may result in leakage.  
# Obtain an AK and SK pair on the management console.  
ak = os.getenv("AccessKeyID")  
sk = os.getenv("SecretAccessKey")  
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from  
# environment variables.  
security_token = os.getenv("SecurityToken")  
# Set server to the endpoint of the region where the bucket is located.  
server = "https://your-endpoint"  
  
# Create an obsClient instance.  
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify  
# security_token when creating an instance.  
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)  
  
# Create an instance of BucketClient.  
bucketClient = obsClient.bucketClient('bucketname')  
# Create a Bucket.  
resp = bucketClient.createBucket()  
if resp.status < 300:
```

```
print('requestId:', resp.requestId)
else:
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
```

NOTE

Except for **ObsClient.listBuckets**, **ObsClient.downloadFile**, **ObsClient.uploadFile**, **ObsClient.createPostSignature**, and **ObsClient.createSignedUrl**, **BucketClient** can implement the same APIs as **ObsClient**, including the same functions and parameters, with the **bucketName** parameter omitted.

6.3 Log Initialization (SDK for Python)

Function

You can enable the SDK log function to record log information generated during API calling into log files for subsequent data analysis or fault location. The procedure is as follows:

1. Find file **log.conf** in the OBS Python SDK. The content format is as follows:

```
[LOGCONF]

#Configure log file dir
LogFileDir      = ./logs

#Configure log file name
LogFileName     = eSDK-OBS-PYTHON.log

#Configure log file size, unit:MB
LogFileSize     = 30

#Configure max log file numbers
LogFileNumber   = 5

#Configure log level for log file (DEBUG | INFO | WARNING | ERROR)
LogFileLevel    = INFO

#Configure whether to print log to console (Yes:1 No:0)
PrintLogToConsole = 0

#Configure log level for console (DEBUG | INFO | WARNING | ERROR)
PrintLogLevel   = WARNING
```

2. Modify parameters in the **log.conf** file as needed.
3. Call **ObsClient.initLog** to enable the logging function.

NOTE

- The logging function is disabled by default. You need to enable it manually.
- For details about SDK logs, see [Log Analysis](#).
- You can change the log file permissions in the system based on your actual needs.

NOTICE

The log module of the OBS Python SDK is thread secure but not process secure. If **ObsClient** is used in multi-process scenarios, you must configure an independent log path for each instance of **ObsClient** to prevent conflicts when multiple processes write logs concurrently.

Method

```
obsClient.initLog(  
    log_config='*** Your Log Configuration Parameters ***',  
    log_name='*** Your Log Name ***'  
)
```

Constructor Parameter Description

Parameter	Type	Mand atory (Yes/N o)	Description
log_config	LogConf	Yes	Log configuration parameter
log_name	str	No	Log name

Table 6-1 LogConf

Parameter	Type	Mandat ory (Yes/No)	Description
config_file	str	Yes	Path to the log configuration file
sec	str	No	Section name in the log configuration file. The default value is LOGCONF .

Code Examples

```
# Import the module.  
from obs import ObsClient  
  
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using  
# hard coding may result in leakage.  
# Obtain an AK and SK pair on the management console.  
ak = os.getenv("AccessKeyID")  
sk = os.getenv("SecretAccessKey")  
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from  
# environment variables.  
security_token = os.getenv("SecurityToken")  
# Set server to the endpoint of the region where the bucket is located.  
server = "https://your-endpoint"  
  
# Create an obsClient instance.  
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify  
# security_token when creating an instance.  
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)  
  
# Import the log module.  
from obs import LogConf  
  
# Specify the path to the log configuration file and initialize logs of ObsClient.  
obsClient.initLog(LogConf('./log.conf'), '*** Your Log Name ***')  
  
# Use ObsClient to access OBS.
```

```
# Disable ObsClient logging.  
obsClient.close()
```

6.4 SDK Common Result Objects (SDK for Python)

Function

Each time an ObsClient related API is called (excluding **ObsClient.createSignedUrl** and **ObsClient.createPostSignature**), an SDK common result object will be returned. You can obtain the HTTP status code from this object to check whether the operation is successful.

Parameter Description

Table 6-2 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

The following is an example of a successful returned result:

```
{  
  'status': 200,  
  'reason': 'OK',  
  'body': {
```

```
...The details are omitted here.  
,  
'requestId': '000001*****B5512',  
'header': [...The header is omitted.]  
}
```

Code Examples

```
try:  
    resp = obsClient.getObject('bucketname', 'objectname', loadStreamInMemory=True)  
  
    if resp.status < 300:  
        print('requestId:', resp.requestId)  
        # Obtain the object content.  
        print('buffer:', resp.body.buffer)  
        print('size:', resp.body.size)  
    else:  
        print('errorCode:', resp.errorCode)  
        print('errorMessage:', resp.errorMessage)  
except:  
    import traceback  
    print(traceback.format_exc())
```

NOTE

In a binary download, if **loadStreamInMemory** is set to **True**, the object content is contained in the **body.buffer** parameter in the returned result.

7 Bucket-Related APIs (SDK for Python)

7.1 Creating a Bucket (SDK for Python)

Function

This API creates an OBS bucket. Buckets are containers for storing objects (files uploaded to OBS) in OBS.

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.
- 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.
- The created 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 an existing bucket name 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.

Method

`ObsClient.createBucket(bucketName, header, location, extensionHeaders)`

Request Parameters

Table 7-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. Default value: None
header	CreateBucketHeader	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the bucket Value range: See Table 7-2. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	Yes if the region where the OBS service resides is not the default region	Explanation: Region where a bucket will be created.
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-2 CreateBucketHeader

Parameter	Type	Mandatory (Yes/No)	Description
aclControl	str	No	Explanation: ACL that can be pre-defined during bucket creation Value range: For details, see HeadPermission . Default value: PRIVATE
storageClass	str	No	Explanation: Bucket storage class that can be pre-defined during bucket creation Value range: See Table 7-4 . Default value: STANDARD

Parameter	Type	Man dator y (Yes/ No)	Description
extensionGrants	list of ExtensionGrant	No	Explanation: Extended permission list that can be specified during bucket creation Value range: See Table 7-5. Default value: None
epid	str	No	Explanation: Enterprise project ID that can be specified during bucket creation. If you have enabled Enterprise Project Management Service (EPS), you can obtain the project ID from the EPS console. Restrictions: The value of epid is a Universally Unique Identifier (UUID). epid is not required if you have not enabled EPS yet. Example: 9892d768-2d13-450f-aac7-ed0e44c2585f Default value: None
isPFS	bool	No	Explanation: Whether a parallel file system is created Value range: • True: A parallel file system is created. • False: A bucket is created. Default value: False

Table 7-3 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 7-4 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 7-5 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeId	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Custom permissions for the bucket Value range: You can select one or more permissions from Table 7-6 . Default value: None

Table 7-6 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Responses

Table 7-7 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-8 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example creates a bucket named **examplebucket** and specifies its location, ACL, storage class, and redundancy type.

```
from obs import CreateBucketHeader, HeadPermission
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Add additional headers to specify a private bucket in the Standard storage class that supports multi-AZ
    # storage.
    header = CreateBucketHeader(aclControl=HeadPermission.PRIVATE, storageClass="STANDARD",
    availableZone="3az")
    # Specify the region where the bucket is to be created. The region must be the same as that in the
    # endpoint passed.
    location = "region"
    bucketName = "examplebucket"
    # Create a bucket.
    resp = obsClient.createBucket(bucketName, header, location)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Create Bucket Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Create Bucket Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Create Bucket Failed')
    print(traceback.format_exc())
```

7.2 Obtaining a Bucket List (SDK for Python)

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 obtain a bucket list, you must have the **obs:bucket:ListAllMyBuckets** permission.

Method

ObsClient.listBuckets(isQueryLocation, extensionHeaders)

Request Parameters

Table 7-9 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
isQueryLocation	bool	No	Explanation: Whether to query the bucket location Value range: - True: The bucket location is queried. - False: The bucket location is not queried. Default value: False
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 7-10 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-11 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-12 GetResult.body

GetResult.body Type	Description
ListBucketsResponse	Explanation: Returned results for obtaining the bucket list. For details, see Table 7-13 .

Table 7-13 ListBucketsResponse

Parameter	Type	Description
buckets	list of Bucket	Explanation: List of buckets Value range: See Table 7-14 .
owner	Owner	Explanation: Bucket owner Value range: See Table 7-15 .

Table 7-14 Bucket

Parameter	Type	Description
name	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. Default value: None
create_date	str	Explanation: Time when the bucket was created. Default value: None

Parameter	Type	Description
location	str	Explanation: Region where a bucket is located
bucket_type	str	Explanation: Bucket type. Value range: - Object bucket: OBJECT - Parallel file system: POSIX Default value: None

Table 7-15 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Code Examples

This example lists all buckets and queries the bucket region.

```
from obs import ObsClient
import os
import traceback
```

```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
# List buckets and set isQueryLocation to True to query the bucket region.
    resp = obsClient.listBuckets(True)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('List Buckets Succeeded')
        print('requestId:', resp.requestId)
        print('name:', resp.body.owner.owner_id)
        print('create_date:', resp.body.owner.owner_name)
        index = 1
        for bucket in resp.body.buckets:
            print('bucket [' + str(index) + ']')
            print('name:', bucket.name)
            print('create_date:', bucket.create_date)
            print('location:', bucket.location)
            index += 1
    else:
        print('List Buckets Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('List Buckets Failed')
    print(traceback.format_exc())
```

7.3 Checking Whether a Bucket Exists (SDK for Python)

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 determine 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).

Method

ObsClient.headBucket(bucketName, extensionHeaders)

Request Parameters

Table 7-16 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Table 7-17 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-18 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example checks whether bucket **examplebucket** exists.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Check whether the bucket exists.
    resp = obsClient.headBucket(bucketName)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Head Bucket Succeeded')
        print('Bucket exists')
    elif resp.status == 404:
        print('Head Bucket Failed')
        print('Bucket does not exist')
    else:
        print('Head Bucket Failed')
        print('status:', resp.status)
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Head Bucket Failed')
    print(traceback.format_exc())
```

7.4 Deleting a Bucket (SDK for Python)

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.

Restrictions

- Only empty buckets can be deleted. An empty bucket means that:
 - The bucket does not contain any object (including noncurrent 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.
- 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).

Method

ObsClient.deleteBucket(bucketName, extensionHeaders)

Request Parameters

Table 7-19 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Table 7-20 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-21 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example deletes bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Delete the bucket.
    resp = obsClient.deleteBucket(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Failed')
    print(traceback.format_exc())
```

7.5 Listing Objects in a Bucket (SDK for Python)

Function

This API lists some or all of the objects in a bucket. You can use parameters such as the prefix, number of returned objects, and start position to list objects that meet specified criteria. 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).

Method

ObsClient.listObjects(bucketName, prefix, marker, max_keys, delimiter, extensionHeaders)

Request Parameters

Table 7-22 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
prefix	str	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 you leave this parameter blank, all objects in the bucket will be returned. Value range: The value must contain 1 to 1,024 characters. Default value: None
marker	str	No	Explanation: Name of the object to start with when listing objects in a bucket. All objects following this object are listed in alphabetical order by object name. Assume that you have the following objects: test/a, test/b, test/c, and test/d. If you specify test/b as the marker, test/c and test/d are returned. Restrictions: This parameter is only used for listing objects with a single version. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
max_keys	int	No	Explanation: The maximum number of objects returned in the response in alphabetical order 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	str	No	Explanation: This parameter is used to group object names. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefix. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefix. 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 CommonPrefix with abcd as the prefix. If only delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefix with abcd as the prefix, and bbcde is grouped separately into another CommonPrefix 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, including the content in subdirectories. In big data scenarios, parallel file systems usually have multiple directory levels and each directory level has a large number of files. In such case, you are advised to configure [delimiter=/] to list the content in the current directory but exclude the content in subdirectories, thereby making the listing more efficient. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 7-23 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-24 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-25 GetResult.body

GetResult.body Type	Description
ListObjectsResponse	Explanation: See Table 7-26 .

Table 7-26 ListObjectsResponse

Parameter	Type	Description
name	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
location	str	Explanation: Region where a bucket is located

Parameter	Type	Description
prefix	str	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 you leave this parameter blank, all objects in the bucket will be returned. Value range: The value must contain 1 to 1,024 characters. Default value: None
marker	str	Explanation: Name of the object to start with when listing objects in a bucket. All objects following this object are listed in alphabetical order by object name. Assume that you have the following objects: test/a, test/b, test/c, and test/d. If you specify test/b as the marker, test/c and test/d are returned. Restrictions: This parameter is only used for listing objects with a single version. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
delimiter	str	Explanation: This parameter is used to group object names. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefix. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefix. 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 commonPrefix with abcd as the prefix. If only delimiter is set to d, objects abcd and abcde are grouped into a commonPrefix with abcd as the prefix, and bbcde is grouped separately into another commonPrefix with bbcd as the prefix. Value range: The value must contain 1 to 1,024 characters. Default value: None
max_keys	int	Explanation: The maximum number of objects returned in the response in alphabetical order 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	Description
is_truncated	bool	Explanation: Whether all results are returned in the response Value range: • True indicates that not all results are returned. • False indicates that all results are returned. Default value: None
next_marker	str	Explanation: Object name to start with upon next request for listing objects Value range: An object name string Default value: None
contents	list of Content	Explanation: List of objects in the bucket Value range: See Table 7-27 .
commonPrefixs	list of CommonPrefix	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: See Table 7-29 .

Table 7-27 Content

Parameter	Type	Description
key	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the ISO8601 format. Example: 2018-01-01T00:00:00.000Z Default value: None
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
size	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
owner	Owner	Explanation: Object owner Value range: See Table 7-28 .
storageClass	str	Explanation: Storage class of the object Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 7-30. Default value: None
isAppendable	bool	Explanation: Whether the object is appendable Value range: True: The object is appendable. False: The object is not appendable. Default value: None

Table 7-28 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 7-29 CommonPrefix

Parameter	Type	Description
prefix	str	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 7-30 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

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 use the folder name as the prefix to list objects in a folder. This example lists objects with **test/** as the prefix in bucket **examplebucket**. A maximum of 100 objects can be returned at a time.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
```

```
# Specify an object prefix.
prefix = 'test/'
# Specify the maximum number of objects to be listed at a time. 100 is used as an example.
max_keys = 100
# List objects in the bucket.
resp = obsClient.listObjects(bucketName, prefix, max_keys=max_keys, encoding_type='url')

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('List Objects Succeeded')
    print('requestId:', resp.requestId)
    print('name:', resp.body.name)
    print('prefix:', resp.body.prefix)
    print('max_keys:', resp.body.max_keys)
    print('is_truncated:', resp.body.is_truncated)
    index = 1
    for content in resp.body.contents:
        print('object [' + str(index) + ']')
        print('key:', content.key)
        print('lastModified:', content.lastModified)
        print('etag:', content.etag)
        print('size:', content.size)
        print('storageClass:', content.storageClass)
        print('owner_id:', content.owner.owner_id)
        print('owner_name:', content.owner.owner_name)
        index += 1
else:
    print('List Objects Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('List Objects Failed')
    print(traceback.format_exc())
```

Sample Code (for Listing over 1,000 Objects)

```
try:
    max_num = 1000
    mark = None
    index = 1
    while True:
        resp = obsClient.listObjects('bucketname', marker=mark, max_keys=max_num, encoding_type='url')
        if resp.status < 300:
            print('requestId:', resp.requestId)
            print('name:', resp.body.name)
            print('prefix:', resp.body.prefix)
            print('max_keys:', resp.body.max_keys)
            print('is_truncated:', resp.body.is_truncated)
            for content in resp.body.contents:
                print('object [' + str(index) + ']')
                print('key:', content.key)
                print('lastModified:', content.lastModified)
                print('etag:', content.etag)
                print('size:', content.size)
                print('storageClass:', content.storageClass)
                print('owner_id:', content.owner.owner_id)
                print('owner_name:', content.owner.owner_name)
                index += 1
            if resp.body.is_truncated is True:
                mark = resp.body.next_marker
            else:
                break
        else:
            print('errorCode:', resp.errorCode)
            print('errorMessage:', resp.errorMessage)
            break
except:
```

```
import traceback
print(traceback.format_exc())
```

7.6 Listing Object Versions in a Bucket (SDK for Python)

Function

This API lists some or all of the object versions in a bucket. You can use parameters such as the prefix, number of returned object versions, and start position to list the object versions that meet specified criteria. 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).

Method

ObsClient.listVersions(bucketName, version, extensionHeaders)

Request Parameters

Table 7-31 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
version	Versions	No	Explanation: Object name to start with when listing object versions in a bucket. All object versions following this parameter are listed in alphabetical order. Restrictions: This parameter is only used for listing objects with multiple versions. Value range: See Table 7-32. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 7-32 Versions

Parameter	Type	Mandatory (Yes/No)	Description
prefix	str	No	Explanation: Name prefix that the objects to be listed must contain Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
key_marker	str	No	Explanation: Object name to start with when listing object versions in a bucket. All object versions following the value specified by this parameter are listed in alphabetical order. Restrictions: This parameter is only used for listing objects with multiple versions. Value range: The value of nextKeyMarker in the response body of the last request Default value: None
max_keys	int or str	No	Explanation: The maximum number of objects returned in the response in alphabetical order 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	str	No	Explanation: This parameter is used to group object names. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one CommonPrefix. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefix. 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 CommonPrefix with abcd as the prefix. If only delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefix with abcd as the prefix, and bbcde is grouped separately into another CommonPrefix 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, including the content in subdirectories. In big data scenarios, parallel file systems usually have multiple directory levels and each directory level has a large number of files. In such case, you are advised to configure [delimiter=/] to list the content in the current directory but exclude the content in subdirectories, thereby making the listing more efficient. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
version_id_marker	str	No	Explanation: Version ID to start with when listing objects in a bucket. All objects are listed in alphabetical order by object name and version ID. This parameter must be used together with key_marker. key_marker specifies the object name, and version_id_marker specifies the version of the specified object. Restrictions: • This parameter is only used for listing objects with multiple versions. • If the value of version_id_marker is not a version ID specified by key_marker, version_id_marker does not take effect. Value range: Object version ID, that is, the value of nextVersionIdMarker in the response body of the last request Default value: None

Responses

Table 7-33 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-34 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-35 GetResult.body

GetResult.body Type	Description
ObjectVersions	Explanation: Response to the request for listing object versions in a bucket

Table 7-36 ObjectVersions

Parameter	Type	Description
head	ObjectVersionHead	Explanation: Response header of the request for listing object versions in a bucket Value range: See Table 7-37 .

Parameter	Type	Description
versions	list of ObjectVersion	Explanation: List of object versions in the bucket Value range: See Table 7-38 .
markers	list of ObjectDeleteMarker	Explanation: List of delete markers in the bucket Value range: See Table 7-39 .
commonPrefixes	list of CommonPrefix	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: See Table 7-41 .

Table 7-37 ObjectVersionHead

Parameter	Type	Description
name	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
location	str	Explanation: Region where a bucket is located

Parameter	Type	Description
delimiter	str	Explanation: This parameter is used to group object names. If a prefix is specified, objects with the same string from the prefix to the first delimiter are grouped into one commonPrefix. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one commonPrefix. 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 commonPrefix with abcd as the prefix. If only delimiter is set to d, objects abcd and abcde are grouped into a commonPrefix with abcd as the prefix, and bbcde is grouped separately into another commonPrefix 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, including the content in subdirectories. In big data scenarios, parallel file systems usually have multiple directory levels and each directory level has a large number of files. In such case, you are advised to configure [delimiter=/] to list the content in the current directory but exclude the content in subdirectories, thereby making the listing more efficient. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
prefix	str	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 you leave this parameter blank, all objects in the bucket will be returned. Value range: The value must contain 1 to 1,024 characters. Default value: None
keyMarker	str	Explanation: Object name to start with for listing object versions Restrictions: This parameter is only used for listing objects with multiple versions. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionIdMarker	str	Explanation: Version ID to start with for listing versioning objects, which is consistent with that set in the request Restrictions: This parameter is only used for listing objects with multiple versions. Value range: The value must contain 32 characters.
nextKeyMarker	str	Explanation: Object name to start with for the next request for listing object versions Value range: An object name string

Parameter	Type	Description
nextVersionIdMarker	str	Explanation: Version ID to start with upon the next request for listing object versions. It is used with the nextKeyMarker parameter. Value range: The value must contain 32 characters.
maxKeys	int	Explanation: The maximum number of objects returned in the response in alphabetical order 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
isTruncated	bool	Explanation: Whether all results are returned in the response Value range: • True indicates that not all results are returned. • False indicates that all results are returned.

Table 7-38 ObjectVersion

Parameter	Type	Description
key	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
versionId	str	Explanation: Object version ID Value range: The value must contain 32 characters. Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the ISO8601 format, for example, 2018-01-01T00:00:00.000Z. Default value: None
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
size	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
owner	Owner	Explanation: Object owner Value range: See Table 7-40 .
storageClass	str	Explanation: Storage class of the object Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 7-42.
isLatest	bool	Explanation: Whether the object is of the latest version Value range: True indicates that the version is the latest. False indicates that the version is not the latest. Default value: None
isAppendable	bool	Explanation: Whether the object is an appendable object Default value: None

Table 7-39 ObjectDeleteMarker

Parameter	Type	Description
key	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	Explanation: Object version ID Value range: The value must contain 32 characters. Default value: None
isLatest	bool	Explanation: Whether the object is of the latest version Value range: • True indicates that the version is the latest. • False indicates that the version is not the latest. Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the ISO8601 format, for example, 2018-01-01T00:00:00.000Z. Default value: None
owner	Owner	Explanation: Object owner Value range: See Table 7-40.

Table 7-40 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 7-41 CommonPrefix

Parameter	Type	Description
prefix	str	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 7-42 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example lists the first 100 objects with **test/** as the prefix in bucket **examplebucket** in alphabetical order.

```
from obs import ObsClient
from obs import Versions
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Specify an object prefix.
    prefix='test/'
    max_keys=100
```

```
# List object versions in a bucket.
resp = obsClient.listVersions(bucketName, version=Versions(prefix, max_keys, encoding_type='url'))
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('List Versions Succeeded')
    print('requestId:', resp.requestId)
    print('name:', resp.body.head.name)
    print('prefix:', resp.body.head.prefix)
    print('maxKeys:', resp.body.head.maxKeys)
    print('isTruncated:', resp.body.head.isTruncated)
    index = 1
    for version in resp.body.versions:
        print('version [' + str(index) + ']')
        print('key:', version.key)
        print('versionId:', version.versionId)
        print('lastModified:', version.lastModified)
        print('etag:', version.etag)
        print('size:', version.size)
        print('storageClass:', version.storageClass)
        print('owner_id:', version.owner.owner_id)
        print('owner_name:', version.owner.owner_name)
        index += 1

    index = 1
    for marker in resp.body.markers:
        print('marker [' + str(index) + ']')
        print('key:', marker.key)
        print('versionId:', marker.versionId)
        print('lastModified:', marker.lastModified)
        print('owner_id:', marker.owner.owner_id)
        print('owner_name:', marker.owner.owner_name)
        index += 1
else:
    print('List Versions Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('List Versions Failed')
    print(traceback.format_exc())
```

7.7 Obtaining Bucket Metadata (SDK for Python)

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).

Method

ObsClient.getBucketMetadata(bucketName, origin, requestHeaders, extensionHeaders)

Request Parameters

Table 7-43 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: You can configure one or more rules and use at most one wildcard character (*) in each rule. If you want to configure multiple rules, separate them using a line breaker. Default value: None
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request. Only CORS requests matching the allowed headers are valid. Restrictions: You can enter multiple allowed headers, with one separated from another using a line break. Each header can contain one wildcard character (*) at most. Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Table 7-44 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-45 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-46 GetResult.body

GetResult.body Type	Description
GetBucketMetadataResponse	Explanation: Response to the request for obtaining bucket metadata

Table 7-47 GetBucketMetadataResponse

Parameter	Type	Description
location	str	Explanation: Region where a bucket is located
storageClass	str	Explanation: Storage class of the bucket. For details, see Table 7-48 . Default value: None
accessControlAllowedOrigin	str	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 the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com . Default value: None
accessControlAllowedHeaders	str	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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlAllow-Methods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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
accessControlExpose-Headers	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlMaxAge	int	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 only one MaxAgeSeconds. Value range: An integer greater than or equal to 0, in seconds Default value: 100
obsVersion	str	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
availableZone	str	Explanation: Data redundancy type that can be specified during bucket creation Restrictions: Multi-AZ redundancy is not available for storage. If the region where the bucket is located does not support multi-AZ storage, single-AZ storage is used by default. Value range: If multi-AZ storage is configured for the bucket, 3az is returned. If single-AZ storage is configured for the bucket, None is returned. Default value: None

Parameter	Type	Description
epid	str	Explanation: Enterprise project ID that can be specified during bucket creation. If you have enabled Enterprise Project Management Service (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

Table 7-48 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example returns the metadata of bucket **examplebucket**. The origin of the cross-origin request is **http://www.a.com** and the HTTP header is **x-obs-header**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Specify the origin (usually a domain name) of the cross-origin request.
    origin='http://www.a.com'
    # Specify the HTTP headers of the cross-origin request.
    requestHeaders='x-obs-header'
    # Obtain the bucket metadata.
    resp = obsClient.getBucketMetadata(bucketName,origin,requestHeaders)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Metadata Succeeded')
        print('requestId:', resp.requestId)
        print('storageClass:', resp.body.storageClass)
        print('accessControlAllowOrigin:', resp.body.accessControlAllowOrigin)
        print('accessControlMaxAge:', resp.body.accessControlMaxAge)
        print('accessControlExposeHeaders:', resp.body.accessControlExposeHeaders)
        print('accessControlAllowMethods:', resp.body.accessControlAllowMethods)
        print('accessControlAllowHeaders:', resp.body.accessControlAllowHeaders)
    else:
        print('Get Bucket Metadata Failed')
        print('requestId:', resp.requestId)
        print('status:', resp.status)
except:
    print('Get Bucket Metadata Failed')
    print(traceback.format_exc())
```

7.8 Obtaining the Region of a Bucket (SDK for Python)

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).

Method

ObsClient.getBucketLocation(bucketName, extensionHeaders)

Request Parameters

Table 7-49 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 7-50 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-51 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-52 GetResult.body

GetResult.body Type	Description
LocationResponse	Explanation: Response to the request for obtaining a bucket location

Table 7-53 LocationResponse

Parameter	Type	Description
location	str	Explanation: Region where a bucket is located

Code Examples

This example returns the region where bucket **examplebucket** is located.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
```

```
bucketName="examplebucket"
# Obtain the bucket region.
resp = obsClient.getBucketLocation(bucketName)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Get Bucket Location Succeeded')
    print('requestId:', resp.requestId)
    print('location:', resp.body.location)
else:
    print('Get Bucket Location Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Location Failed')
    print(traceback.format_exc())
```

7.9 Obtaining Storage Information of a Bucket (SDK for Python)

Function

This API returns the storage information of 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).

Method

ObsClient.getBucketStorageInfo(bucketName, extensionHeaders)

Request Parameters

Table 7-54 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Table 7-55 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-56 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-57 GetResult.body

GetResult.body Type	Description
GetBucketStorageInfoResponse	Explanation: Response result of the request for obtaining storage information about a bucket

Table 7-58 GetBucketStorageInfoResponse

Parameter	Type	Description
size	long	Explanation: Bucket size Default value: None
objectNumber	int	Explanation: Number of objects in the bucket Default value: None

Code Examples

This example returns the storage information of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain bucket storage information.
    resp = obsClient.getBucketStorageInfo(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket StorageInfo Succeeded')
        print('requestId:', resp.requestId)
        print('size:', resp.body.size)
        print('objectNumber:', resp.body.objectNumber)
    else:
```

```
print('Get Bucket StorageInfo Failed')
print('requestId:', resp.requestId)
print('errorCode:', resp.errorCode)
print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket StorageInfo Failed')
    print(traceback.format_exc())
```

7.10 Configuring a Storage Quota (SDK for Python)

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. After the storage quota has been 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, to upload new objects, you must delete some existing objects to make the used space below the quota limit.

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).

Method

`ObsClient.setBucketQuota(bucketName, quota, extensionHeaders)`

Request Parameters

Table 7-59 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
quota	int	Yes	Explanation: Bucket quota Value range: An integer greater than or equal to 0, in bytes Default value: 0, indicating that there is no limit on the bucket quota.

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 7-60 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-61 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures a 1 GB quota for bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Specify a 1 GB quota (measured in bytes) for the bucket.
    quota = 1024 * 1024 * 1024
    # Configure a quota for the bucket.
    resp = obsClient.setBucketQuota(bucketName, quota)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Quota Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Quota Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Quota Failed')
    print(traceback.format_exc())
```

7.11 Obtaining a Bucket Storage Quota (SDK for Python)

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).

Method

`ObsClient.getBucketQuota(bucketName, extensionHeaders)`

Request Parameters

Table 7-62 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Table 7-63 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 7-64 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-65 GetResult.body

GetResult.body Type	Description
GetBucketQuotaResponse	Explanation: Response result of the request for obtaining the storage capacity quota of a bucket

Table 7-66 GetBucketQuotaResponse

Parameter	Type	Description
quota	int	Explanation: Bucket quota Value range: An integer greater than or equal to 0, in bytes Default value: 0 , indicating that there is no limit on the bucket quota.

Code Examples

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

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the bucket quota.
    resp = obsClient.getBucketQuota(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Quota Succeeded')
        print('requestId:', resp.requestId)
        print('quota:', resp.body.quota)
    else:
        print('Get Bucket Quota Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Quota Failed')
    print(traceback.format_exc())
```

7.12 Configuring a Storage Class for a Bucket (SDK for Python)

Function

This API configures a storage class for a bucket. If you do not specify a storage class when uploading or copying an object, or initiating a multipart upload, the object inherits the bucket's storage class.

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).

Method

`ObsClient.setBucketStoragePolicy(bucketName, storageClass, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
storageClass	str	Yes	Explanation: Storage class of the bucket Value range: See Table 7-67. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-67 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-68 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None

Parameter	Type	Description
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example sets the storage class of bucket **examplebucket** to **Standard**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
```

```
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Set the storage class of the bucket to Standard.
    storageClass = 'STANDARD'
    # Configure the storage class for the bucket.
    resp = obsClient.setBucketStoragePolicy(bucketName, storageClass)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket StoragePolicy Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket StoragePolicy Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket StoragePolicy Failed')
    print(traceback.format_exc())
```

7.13 Obtaining the Storage Class of a Bucket (SDK for Python)

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).

Method

ObsClient.getBucketStoragePolicy(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-69 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
GetBucketStoragePolicyResponse	Explanation: Response to the request for obtaining the storage class of a bucket

Table 7-70 GetBucketStoragePolicyResponse

Parameter	Type	Description
storageClass	str	Explanation: Storage class of the bucket

Table 7-71 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example returns the storage class of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
```



```
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the bucket's storage class.
    resp = obsClient.getBucketStoragePolicy(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket StoragePolicy Succeeded')
        print('requestId:', resp.requestId)
        print('storageClass:', resp.body.storageClass)
    else:
        print('Get Bucket StoragePolicy Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket StoragePolicy Failed')
    print(traceback.format_exc())
```

7.14 Configuring a Bucket ACL (SDK for Python)

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.

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).

Method

ObsClient.setBucketAcl(bucketName, acl, aclControl, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
acl	ACL	No	Explanation: Bucket ACL. Value range: See Table 7-72 . Default value: None NOTE acl and aclControl are mutually exclusive.
aclControl	str	No	Explanation: Pre-defined ACL. Value range: For details, see HeadPermission . Default value: None NOTE acl and aclControl are mutually exclusive.
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-72 ACL

Parameter	Type	Mandatory (Yes/No)	Description
owner	Owner	Yes if used as a request parameter	Explanation: Bucket owner information. For details, see Table 7-75. Restrictions: Owner and Grants must be used together and they cannot be used with aclControl. Default value: None
grants	list of Grant	Yes if used as a request parameter	Explanation: List of grantees' permission information. For details, see Table 7-76. Default value: None
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-73 Permission

Constant	Description
READ	Read permission 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.

Constant	Description
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Table 7-74 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 7-75 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 7-76 Grant

Parameter	Type	Mandatory (Yes/No)	Description
grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 7-77 .
permission	str	Yes if used as a request parameter	Explanation: Granted permission Value range: You can select one or more permissions from Table 7-73 . Default value: None
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-77 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
grantee_id	str	Yes if the parameter is used as a request parameter and group is left blank	Explanation: Account (domain) ID of the grantee. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
grantee_name	str	No if used as a request parameter	Explanation: Username of the grantee. For details, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Restrictions: • Cannot contain full-width characters. • Starts with a letter. • Contains 6 to 32 characters. • Contains only letters, digits, hyphens (-), and underscores (_). Default value: None
group	str	Yes if the parameter is used as a request parameter and grantee_id is left blank	Explanation: Authorized user group Value range: See Table 7-78 . Default value: None

 NOTE

The authorized entity can be an individual user or a user group. **grantee_id** and **grantee_name** must be used together and they cannot be used with **group**.

Table 7-78 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-79 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example sets the ACL permission control for bucket **examplebucket** to public read for all users and to read and write for an IAM user (**userid**).

```
from obs import ObsClient, HeadPermission
from obs import ACL
from obs import Owner
from obs import Grantee
from obs import Grant
from obs import Group
from obs import Permission
import os
```

```
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the account ID of the bucket owner (ownerid as an example).
    owner_id = 'ownerid'
    owner = Owner(owner_id=owner_id)
    # Specify the IAM user ID (userid).
    grantee1 = Grantee(grantee_id='userid')
    # Specify all users.
    grantee2 = Grantee(group=Group.ALL_USERS)
    # Grant the read and write permissions to the specified IAM user.
    grant1 = Grant(grantee=grantee1, permission=Permission.READ)
    grant2 = Grant(grantee=grantee1, permission=Permission.WRITE)
    # Grant the read permission to all users.
    grant3 = Grant(grantee=grantee2, permission=Permission.READ)
    # Set the ACL permission control for bucket examplebucket to public read for all users and to read and
    # write for the IAM user.
    acl = ACL(owner=owner, grants=[grant1, grant2, grant3])
    bucketName = "examplebucket"
    # Set the bucket ACL.
    resp = obsClient.setBucketAcl(bucketName, acl)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Acl Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Acl Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Acl Failed')
    print(traceback.format_exc())
```

7.15 Obtaining a Bucket ACL (SDK for Python)

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.

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).

Method

ObsClient.getBucketAcl(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-80 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
ACL	Explanation: ACL configuration of the bucket.

Table 7-81 ACL

Parameter	Type	Mandatory (Yes/No)	Description
owner	Owner	Yes if used as a request parameter	Explanation: Bucket owner. For details, see Table 7-82 .
grants	list of Grant	Yes if used as a request parameter	Explanation: Grantees' permission information. For details, see Table 7-83 .
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-82 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 7-83 Grant

Parameter	Type	Mandatory (Yes/No)	Description
grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 7-84 . Default value: None
permission	str	Yes if used as a request parameter	Explanation: Granted permission. For details, see Table 7-85 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-84 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
grantee_id	str	Yes if the parameter is used as a request parameter and group is left blank	Explanation: Account (domain) ID of the grantee. Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
grantee_name	str	No if used as a request parameter	Explanation: Username of the grantee. Restrictions: • Cannot contain full-width characters. • Starts with a letter. • Contains 6 to 32 characters. • Contains only letters, digits, hyphens (-), and underscores (_). Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
group	str	Yes if the parameter is used as a request parameter and grantee_id is left blank	Explanation: Authorized user group Value range: See Table 7-86 . Default value: None

Table 7-85 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

 NOTE

The authorized entity can be an individual user or a user group. **grantee_id** and **grantee_name** must be used together and they cannot be used with **group**.

Table 7-86 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Code Examples

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

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the bucket ACL.
    resp = obsClient.getBucketAcl(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Acl Succeeded')
        print('requestId:', resp.requestId)
        print('owner_id:', resp.body.owner.owner_id)
        print('owner_name:', resp.body.owner.owner_name)
        index = 1
        for grant in resp.body.grants:
            print('grant [' + str(index) + ']')
            print('grant_id:', grant.grantee.grantee_id)
            print('grant_name:', grant.grantee.grantee_name)
            print('group:', grant.grantee.group)
            print('permission:', grant.permission)
            index += 1
    else:
        print('Get Bucket Acl Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Acl Failed')
    print(traceback.format_exc())
```

7.16 Configuring Logging for a Bucket (SDK for Python)

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 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 configure access to log files and the name prefix of 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.
- 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).

Method

`ObsClient.setBucketLogging(bucketName, logstatus, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
logstatus	Logging	No	Explanation: Logging configuration information For details, see Table 7-87. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 7-87 Logging

Parameter	Type	Mandatory (Yes/No)	Description
targetBucket	str	No	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
targetPrefix	str	No	Explanation: Name prefix for log files stored in the log storage bucket Value range: The value must contain 1 to 1,024 characters. Default value: None
targetGrants	list of Grant	No	Explanation: Permission information list of grantees, which defines grantees and their permissions for log files. For details, see Table 7-88. Default value: None
agency	str	Yes when configuring bucket logging	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. 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. Default value: None

Table 7-88 Grant

Parameter	Type	Mandatory (Yes/No)	Description
grantee	Grantee	Yes	Explanation: Grantee information. For details, see Table 7-89 .
permission	str	Yes	Explanation: Granted permission Value range: See Table 7-90 . Default value: None
delivered	bool	No	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-89 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
grantee_id	str	Yes if group is left blank.	Explanation: Account (domain) ID of the grantee. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
grantee_name	str	No	Explanation: Username of the grantee. Restrictions: - Starts with a letter. Contains 6 to 32 characters. - Contains only letters, digits, hyphens (-), and underscores (_). Default value: None
group	str	Yes if grantee_id is left blank.	Explanation: Authorized user group Value range: See Table 7-91 . Default value: None

 NOTE

The authorized entity can be an individual user or a user group. **grantee_id** and **grantee_name** must be used together and they cannot be used with **group**.

Table 7-90 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.

Constant	Description
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Table 7-91 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-92 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures logging for bucket **examplebucket** and store log files in bucket **targetbucket**.

```
from obs import ObsClient
from obs import Grantee
from obs import Grant
from obs import Logging
from obs import Permission
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
```

```
try:
    # Authorize the account ID (domain_id) of a user to read the bucket logs.
    grantee1 = Grantee(grantee_id='domain_id')
    grant1 = Grant(grantee=grantee1, permission=Permission.READ)
    grantList = [grant1]
    # Specify a bucket name for storing generated log files.
    targetBucket = "targetBucket"
    # Specify an object prefix, indicating the path for storing log files.
    targetPrefix = 'test/'
    # Specify an agency name.
    agency = 'your agency'
    # Specify the logging configuration information.
    logstatus = Logging(targetBucket=targetBucket, targetPrefix=targetPrefix, targetGrants=grantList,
agency=agency)
    bucketName = "examplebucket"
    # Configure logging for the bucket.
    resp = obsClient.setBucketLogging(bucketName=bucketName , logstatus=logstatus )

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Logging Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Logging Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Logging Failed')
    print(traceback.format_exc())
```

7.17 Obtaining the Logging Configuration of a Bucket (SDK for Python)

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).

Method

ObsClient.getBucketLogging(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-93 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
Logging	Explanation: Logging configuration information

Table 7-94 Logging

Parameter	Type	Mandatory (Yes/No)	Description
targetBucket	str	No if used as a request parameter	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
targetPrefix	str	No if used as a request parameter	Explanation: Name prefix for log files stored in the log storage bucket Value range: The value must contain 1 to 1,024 characters. Default value: None
targetGrants	list of Grant	No if used as a request parameter	Explanation: Permission information list of grantees, which defines grantees and their permissions for log files. For details, see Table 7-95. Default value: None
agency	str	Yes if the parameter is in a request to enable bucket logging	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. 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. Default value: None

Table 7-95 Grant

Parameter	Type	Description
grantee	Grantee	Explanation: Grantee information. For details, see Table 7-96 . Default value: None
permission	str	Explanation: Granted permission Value range: See Table 7-98 . Default value: None
delivered	bool	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 7-96 Grantee

Parameter	Type	Description
grantee_id	str	Explanation: Account (domain) ID of the grantee. Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Parameter	Type	Description
grantee_name	str	Explanation: Account name of the grantee. Restrictions: - Starts with a letter. Contains 6 to 32 characters. - Contains only letters, digits, hyphens (-), and underscores (_). Default value: None
group	str	Explanation: Authorized user group. For details, see Table 7-97 . Default value: None

 NOTE

The authorized entity can be an individual user or a user group. **grantee_id** and **grantee_name** must be used together and they cannot be used with **group**.

Table 7-97 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Table 7-98 Permission

Constant	Description
READ	Read permission 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.

Constant	Description
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Code Examples

This example returns the logging configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the logging configuration of the bucket.
```

```
resp = obsClient.getBucketLogging(bucketName)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Get Bucket Logging Succeeded')
    print('requestId:', resp.requestId)
    print('targetBucket:', resp.body.targetBucket)
    print('targetPrefix:', resp.body.targetPrefix)

    index = 1
    for grant in resp.body.targetGrants:
        print('grant [' + str(index) + ']')
        print('grant_id:', grant.grantee.grantee_id)
        print('grant_name:', grant.grantee.grantee_name)
        print('group:', grant.grantee.group)
        print('permission:', grant.permission)
        index += 1
    else:
        print('Get Bucket Logging Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Logging Failed')
    print(traceback.format_exc())
```

7.18 Configuring a Bucket Policy (SDK for Python)

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.

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permissions 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 can configure required bucket policies.

This API configures a policy for a bucket.

Restrictions

- Due to data caching, after a bucket policy is configured, it takes 5 minutes at most 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).

Method

`ObsClient.setBucketPolicy(bucketName, policyJSON, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandator y (Yes/No)	Description
policyJSON	str	Yes	Explanation: Policy information in JSON format. Restrictions: The bucket name contained in the Resource parameter of the policy must be the one specified for the current bucket policy. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-99 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None

Parameter	Type	Description
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures a policy for bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Specify the bucket policy. The following example policy grants all operation permissions on
    # examplebucket (including the bucket and its objects) to user xxx (user ID) under tenant xxx (domain ID).
    policyJSON = '{"Statement":[{"Sid":"ExampleStatementID1","Effect":"Allow","Principal":{"ID":["domain/*****.user/*****"]},"Action":["*"],"Resource":["examplebucket/*","examplebucket"]}]}
    # Configure the bucket policy.
    resp = obsClient.setBucketPolicy(bucketName, policyJSON)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Policy Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Policy Failed')
        print('requestId:', resp.requestId)
```

```
print('errorCode:', resp.errorCode)
print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Policy Failed')
    print(traceback.format_exc())
```

7.19 Obtaining the Policy of a Bucket (SDK for Python)

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.

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permissions 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 can configure required bucket policies.

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).

Method

ObsClient.getBucketPolicy(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-100 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
Policy	Explanation: Bucket policy

Table 7-101 Policy

Parameter	Type	Description
policyJSON	str	Explanation: Policy information in JSON format Restrictions: The bucket name contained in the Resource parameter of the policy must be the one specified for the current bucket policy. Default value: None

Code Examples

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

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the bucket policy.
    resp = obsClient.getBucketPolicy(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Policy Succeeded')
        print('requestId:', resp.requestId)
        print('policyJSON:', resp.body.policyJSON)
    else:
        print('Get Bucket Policy Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Policy Failed')
    print(traceback.format_exc())
```


7.20 Deleting a Bucket Policy (SDK for Python)

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.

A bucket policy is applied to a configured bucket and the objects in it. You can use a bucket policy to grant permissions 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 can configure required bucket policies.

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).

Method

`ObsClient.deleteBucketPolicy(bucketName, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-102 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

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

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Delete the bucket policy.
    resp = obsClient.deleteBucketPolicy(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Policy Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Policy Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Policy Failed')
    print(traceback.format_exc())
```

7.21 Configuring Lifecycle Rules for a Bucket (SDK for Python)

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes.

This API configures lifecycle rules for a bucket.

NOTE

- Expired objects will be permanently deleted and cannot be recovered.
- Multi-AZ redundancy is not available for storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the storage class based on a lifecycle rule.
- The minimum storage duration is 30 days for storage, and 90 days for storage. After an object is transitioned to the 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 about 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).

Method

`ObsClient.setBucketLifecycle(bucketName, lifecycle, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-)

Parameter	Type	Mandatory (Yes/No)	Description
			adjacent to each other, for example, my-.bucket or my.-bucket. If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
lifecycle	Lifecycle	Yes	Explanation: List of lifecycle rules. For details, see Table 7-103. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 7-103 Lifecycle

Parameter	Type	Mandatory (Yes/No)	Description
rule	list of Rule	Yes	Explanation: List of lifecycle rules. For details, see Table 7-104 . Default value: None

Table 7-104 Rule

Parameter	Type	Mandatory (Yes/No)	Description
id	str	No	Explanation: Lifecycle rule ID Value range: The value must contain 1 to 255 characters. Default value: None
prefix	str	Yes	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 only. If you set Prefix to logs/ , the rule applies to the three objects with name starting with logs/ . If you leave Prefix blank, the rule applies to all objects in the bucket. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
status	str	Yes	Explanation: Whether the rule is enabled Value range: - Enabled - Disabled Default value: None
transition	Transition or list of Transition	No	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 7-105 . Restrictions: This parameter applies only to the current object version. Default value: None
expiration	Expiration	No	Explanation: Object expiration time. For details, see Table 7-106 . Restrictions: This parameter applies only to the current object version. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
noncurrentVersion-Transition	NoncurrentVersionTransition or list of NoncurrentVersionTransition	No	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 7-109. Restrictions: • This parameter applies only to noncurrent object versions. • Versioning is enabled (or suspended after being enabled) for the bucket. • This parameter is not available for parallel file systems. Default value: None
noncurrentVersionExpiration	NoncurrentVersionExpiration	No	Explanation: Expiration time of noncurrent object versions. For details, see Table 7-110. Restrictions: • This parameter applies only to noncurrent object versions. • Versioning is enabled (or suspended after being enabled) for the bucket. CAUTION This parameter is not available for parallel file systems. Default value: None

Table 7-105 Transition

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	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 storage class are supported. To transition objects from to Standard, you must manually do it. – Only transitions from the Standard or storage class to the storage class are supported. To transition objects from to Standard or , you must restore the objects first and then manually transition their storage class. – Multi-AZ redundancy is not available for storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the storage class based on a lifecycle rule. Value range: See Table 7-107. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
date	str or DateTime	Yes if the 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. For example, DateTime(year=2023, month=9, day=12) Default value: None
days	int	Yes if the parameter is used as a request parameter and date is absent	Explanation: Number of days (since the last update was made to the object) after which the lifecycle rule takes effect Restrictions: This parameter applies only to the current object version. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-106 Expiration

Parameter	Type	Mandatory (Yes/No)	Description
date	str or DateTime	Yes if the 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 deleted. For example, DateTime(year=2023, month=9, day=12) Default value: None
days	int	Yes if the parameter is used as a request parameter and date is absent	Explanation: Number of days (since the last update was made to the object) after which the lifecycle rule takes effect Restrictions: This parameter applies only to the current object version. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-107 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 7-108 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0

Parameter	Type	Description
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

Table 7-109 NoncurrentVersionTransition

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	Yes if used as a request parameter	Explanation: Storage class of noncurrent 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 storage class are supported. To transition objects from to Standard, you must manually do it. – Only transitions from the Standard or storage class to the storage class are supported. To transition objects from to Standard or , you must restore the objects first and then manually transition their storage class. – Multi-AZ redundancy is not available for storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the storage class based on a lifecycle rule. Value range: See Table 7-107. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
noncurrentDays	int	Yes if used as a request parameter	Explanation: Number of days an object is noncurrent before the specified rule takes effect Restrictions: This parameter applies only to noncurrent object versions. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-110 NoncurrentVersionExpiration

Parameter	Type	Mandatory (Yes/No)	Description
noncurrentDays	int	Yes if used as a request parameter	Explanation: Number of days an object is noncurrent before the specified rule takes effect Restrictions: This parameter applies only to noncurrent object versions. Value range: An integer greater than or equal to 0, in days Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-111 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures a lifecycle rule for bucket **examplebucket**.

```
from obs import ObsClient
from obs import Expiration, NoncurrentVersionExpiration
from obs import DateTime
from obs import Rule
from obs import Lifecycle
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
```

```
# Configure a lifecycle rule for deleting objects with prefix1 as the prefix after 60 days since the last
update.
rule1 = Rule(id='rule1', prefix='prefix1', status='Enabled', expiration=Expiration(days=60))
# Configure a lifecycle rule for deleting objects with prefix2 as the prefix on December 31, 2023.
rule2 = Rule(id='rule2', prefix='prefix2', status='Enabled', expiration=Expiration(date=DateTime(2023, 12,
31)))
lifecycle = Lifecycle(rule=[rule1, rule2])
bucketName="examplebucket"
# Configures the lifecycle rules for the bucket.
resp = obsClient.setBucketLifecycle(bucketName, lifecycle)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Set Bucket Lifecycle Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Set Bucket Lifecycle Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Lifecycle Failed')
    print(traceback.format_exc())
```

7.22 Obtaining the Lifecycle Configuration of a Bucket (SDK for Python)

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes.

This API returns the lifecycle rules of a bucket.

Restrictions

- To obtain the lifecycle configuration 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).

Method

ObsClient.getBucketLifecycle(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-112 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
LifecycleResponse	Explanation: Response to the request for obtaining the lifecycle configuration information of a bucket Default value: None

Table 7-113 LifecycleResponse

Parameter	Type	Description
lifecycleConfig	Lifecycle	Explanation: Bucket lifecycle configuration. For details, see Table 7-114 . Default value: None

Table 7-114 Lifecycle

Parameter	Type	Mandatory (Yes/No)	Description
rule	list of Rule	Yes if used as a request parameter	Explanation: List of lifecycle rules. For details, see Table 7-115 . Default value: None

Table 7-115 Rule

Parameter	Type	Mandatory (Yes/No)	Description
id	str	No if used as a request parameter	Explanation: Lifecycle rule ID Value range: The value must contain 1 to 255 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
prefix	str	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 only. If you set Prefix to logs/, the rule applies to the three objects with name starting with logs/. If you leave Prefix blank, the rule applies to all objects in the bucket. Value range: The value must contain 1 to 1,024 characters. Default value: None
status	str	Yes if used as a request parameter	Explanation: Whether the rule is enabled Value range: • Enabled • Disabled Default value: None
transition	Transition or list of Transition	No if used as a request parameter	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 7-116. Restrictions: This parameter applies only to the current object version. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
expiration	Expiration	No if used as a request parameter	Explanation: Object expiration time. For details, see Table 7-118. Restrictions: This parameter applies only to the current object version. Default value: None
noncurrentVersion-Transition	NoncurrentVersion Transition or list of NoncurrentVersion Transition	No if used as a request parameter	Explanation: Policies for storage class transition, including transition time and the storage class after transition. For details, see Table 7-120. Restrictions: - This parameter applies only to noncurrent object versions. - Versioning is enabled (or suspended after being enabled) for the bucket. - This parameter is not available for parallel file systems. Default value: None
noncurrentVersionExpiration	NoncurrentVersion Expiration	No if used as a request parameter	Explanation: Expiration time of noncurrent object versions. For details, see Table 7-121. Restrictions: - This parameter applies only to noncurrent object versions. - Versioning is enabled (or suspended after being enabled) for the bucket. - This parameter is not available for parallel file systems. Default value: None

Table 7-116 Transition

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	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 storage class are supported. To transition objects from to Standard, you must manually do it. – Only transitions from the Standard or storage class to the storage class are supported. To transition objects from to Standard or , you must restore the objects first and then manually transition their storage class. – Multi-AZ redundancy is not available for storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the storage class based on a lifecycle rule. Value range: See Table 7-117. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
date	str or DateTime	Yes if the parameter is used as a request parameter and days is absent	Explanation: Date when an object will be transitioned. 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. For example, DateTime(year=2023, month=9, day=12) Default value: None
days	int	Yes if the parameter is used as a request parameter and date is absent	Explanation: Number of days (since the last update was made to the object) after which the lifecycle rule takes effect Restrictions: This parameter applies only to the current object version. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-117 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 7-118 Expiration

Parameter	Type	Mandatory (Yes/No)	Description
date	str or DateTime	Yes if the 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 deleted. For example, DateTime(year=2023, month=9, day=12) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
days	int	Yes if the parameter is used as a request parameter and date is absent	Explanation: Number of days (since the last update was made to the object) after which the lifecycle rule takes effect Restrictions: This parameter applies only to the current object version. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-119 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0

Parameter	Type	Description
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

Table 7-120 NoncurrentVersionTransition

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	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 storage class are supported. To transition objects from to Standard, you must manually do it. – Only transitions from the Standard or storage class to the storage class are supported. To transition objects from to Standard or , you must restore the objects first and then manually transition their storage class. – Multi-AZ redundancy is not available for storage. For this reason, buckets or objects with multi-AZ redundancy cannot be transitioned to the storage class based on a lifecycle rule. Value range: See Table 7-117. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
noncurrentDays	int	Yes if used as a request parameter	Explanation: Number of days an object is noncurrent before the specified rule takes effect Restrictions: This parameter applies only to noncurrent object versions. Value range: An integer greater than or equal to 0, in days Default value: None

Table 7-121 NoncurrentVersionExpiration

Parameter	Type	Mandatory (Yes/No)	Description
noncurrentDays	int	Yes if used as a request parameter	Explanation: Number of days an object is noncurrent before the specified rule takes effect Restrictions: This parameter applies only to noncurrent object versions. Value range: An integer greater than or equal to 0, in days Default value: None

Code Examples

This example returns the lifecycle configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
```

```
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the lifecycle configuration of the bucket.
    resp = obsClient.getBucketLifecycle(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Lifecycle Succeeded')
        print('requestId:', resp.requestId)
        index = 1
        for rule in resp.body.lifecycleConfig.rule:
            print('rule [' + str(index) + ']')
            print('id:', rule.id)
            print('prefix:', rule.prefix)
            print('status:', rule.status)
            if rule.expiration:
                print('days:', rule.expiration.days)
                print('date:', rule.expiration.date)
            if rule.noncurrentVersionExpiration:
                print('noncurrentDays:', rule.noncurrentVersionExpiration.noncurrentDays)
            index += 1
    else:
        print('Get Bucket Lifecycle Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Lifecycle Failed')
    print(traceback.format_exc())
```

7.23 Deleting the Lifecycle Configuration of a Bucket (SDK for Python)

Function

You can configure lifecycle rules to periodically delete objects or transition objects between storage classes.

This API deletes the lifecycle configuration of a bucket.

Restrictions

- To delete the lifecycle configuration 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).

Method

`ObsClient.deleteBucketLifecycle(bucketName, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-122 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example deletes all lifecycle configurations of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Delete lifecycle configurations of the bucket.
    resp = obsClient.deleteBucketLifecycle(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Lifecycle Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Lifecycle Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Lifecycle Failed')
    print(traceback.format_exc())
```

7.24 Configuring Static Website Hosting for a Bucket (SDK for Python)

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 you use 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).

Method

ObsClient.setBucketWebsite(bucketName, website, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
website	WebsiteConfiguration	Yes	Explanation: Input parameters for configuring static website hosting. For details, see Table 7-123 . Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-123 WebsiteConfiguration

Parameter	Type	Mandatory (Yes/No)	Description
redirectAllRequestTo	RedirectAllRequestTo	No if used as a request parameter	Explanation: Redirection rules for all requests. For details, see Table 7-124 . Default value: None
indexDocument	IndexDocument	No if used as a request parameter	Explanation: Default page configuration. For details, see Table 7-125 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
errorDocument	ErrorDocument	No if used as a request parameter	Explanation: Error page configuration. For details, see Table 7-126 . Default value: None
routingRules	list of RoutingRule	No if used as a request parameter	Explanation: List of routing rules. For details, see Table 7-127 . Default value: None

 NOTE

- **errorDocument**, **indexDocument**, and **routingRules** must be used together and they cannot be used with **redirectAllRequestsTo**.
- When **errorDocument**, **indexDocument**, and **routingRules** are used together, **routingRules** can be left blank.
- You must specify either the combo of fields **ErrorDocument**, **IndexDocument**, and **RoutingRules**, or the **RedirectAllRequestsTo** field.

Table 7-124 RedirectAllRequestTo

Parameter	Type	Mandatory (Yes/No)	Description
hostName	str	Yes if used as a request parameter	Explanation: Host name used for redirection, for example, www.example.com Restrictions: The host name must comply with the host name rules. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
protocol	str	No if used as a request parameter	Explanation: Protocol used for redirection Value range: • http • https Default value: None

Table 7-125 IndexDocument

Parameter	Type	Mandatory (Yes/No)	Description
suffix	str	Yes if used as a request parameter	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 . Value range: This parameter can neither be left blank nor contain slashes (/). Default value: None

Table 7-126 ErrorDocument

Parameter	Type	Mandatory (Yes/No)	Description
key	str	No if used as a request parameter	Explanation: Object name to use when a 4XX error occurs. This parameter specifies the webpage to display when an error occurs. Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 7-127 RoutingRule

Parameter	Type	Mandatory (Yes/No)	Description
condition	Condition	No if used as a request parameter	Explanation: Conditions that must be met for the specified redirect to apply Value range: See Table 7-128 . Default value: None
redirect	Redirect	Yes if used as a request parameter	Explanation: Details about the redirection. For details, see Table 7-129 . Default value: None

Table 7-128 Condition

Parameter	Type	Mandatory (Yes/No)	Description
keyPrefixEquals	str	No if used as a request parameter	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 httpErrorReturnedEquals . Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
httpErrorCodeReturnedEquals	int	No if used as a request parameter	Explanation: HTTP error code for the redirection to take effect. If there is an error, and the 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. Default value: None

Table 7-129 Redirect

Parameter	Type	Mandatory (Yes/No)	Description
protocol	str	No if used as a request parameter	Explanation: Protocol used for redirection Value range: • http • https Default value: None
hostName	str	No if used as a request parameter	Explanation: Host name used for redirection Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
replaceKeyPrefix-With	str	No if used as a request parameter	Explanation: Object name prefix used in the redirection request Value range: The value must contain 1 to 1,024 characters. Default value: None
replaceKeyWith	str	No if used as a request parameter	Explanation: Object name used in the redirection request Restrictions: This parameter cannot be used together with replaceKeyPrefix-With . Value range: The value must contain 1 to 1,024 characters. Default value: None
httpRedirectCode	int	No if used as a request parameter	Explanation: HTTP status code in the response to the redirect request. Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-130 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures static website hosting for bucket **examplebucket**.

```
from obs import ObsClient
from obs import WebsiteConfiguration
from obs import IndexDocument
from obs import ErrorDocument
from obs import RoutingRule
from obs import Condition
from obs import Redirect
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
```



```
try:
    # Specify an error page when a 4XX error occurs.
    errorDocument = ErrorDocument(key='error.html')
    # Specify a default page.
    indexDocument = IndexDocument(suffix='index.html')
    # Specify a rule for redirecting requests to NotFound.html if the status code is 404.
    routingRule1 = RoutingRule(condition=Condition(httpErrorCodeReturnedEquals=404),
                               redirect=Redirect(protocol='http', replaceKeyWith='NotFound.html'))
    # Configure the redirection rules in list format. Multiple rules can be configured.
    routingRules = [routingRule1]
    bucketName = "examplebucket"
    # Configure static website hosting for the bucket.
    resp = obsClient.setBucketWebsite(bucketName,
                                     WebsiteConfiguration(errorDocument=errorDocument,
                                                           indexDocument=indexDocument,
                                                           routingRules=routingRules))

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Website Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Website Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Website Failed')
    print(traceback.format_exc())
```

7.25 Obtaining Static Website Hosting for a Bucket (SDK for Python)

Function

You can host static website resources such as HTML web pages, flash files, as well as 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 configuration of the bucket.

Restrictions

- To obtain the static website hosting configuration 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).

Method

ObsClient.getBucketWebsite(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-131 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
WebsiteConfiguration	Explanation: Website hosting settings of a bucket

Table 7-132 WebsiteConfiguration

Parameter	Type	Mandatory (Yes/No)	Description
redirectAllRequest-To	RedirectAllRequestTo	No if used as a request parameter	Explanation: Redirection rules for all requests. For details, see Table 7-133 .
indexDocument	IndexDocument	No if used as a request parameter	Explanation: Default page configuration. For details, see Table 7-134 .
errorDocument	ErrorDocument	No if used as a request parameter	Explanation: Error page configuration. For details, see Table 7-135 .
routingRules	list of RoutingRule	No if used as a request parameter	Explanation: List of routing rules. For details, see Table 7-136 .

 NOTE

- **errorDocument**, **indexDocument**, and **routingRules** must be used together and they cannot be used with **redirectAllRequestsTo**.
- When **errorDocument**, **indexDocument**, and **routingRules** are used together, **routingRules** can be left blank.
- You must specify either the combo of fields **ErrorDocument**, **IndexDocument**, and **RoutingRules**, or the **RedirectAllRequestsTo** field.

Table 7-133 RedirectAllRequestTo

Parameter	Type	Mandatory (Yes/No)	Description
hostName	str	Yes if used as a request parameter	Explanation: Host name used for redirection, for example, www.example.com Restrictions: The host name must comply with the host name rules. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
protocol	str	No if used as a request parameter	Explanation: Protocol used for redirection Value range: • http • https Default value: None

Table 7-134 IndexDocument

Parameter	Type	Mandatory (Yes/No)	Description
suffix	str	Yes if used as a request parameter	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 . Value range: This parameter can neither be left blank nor contain slashes (/). Default value: None

Table 7-135 ErrorDocument

Parameter	Type	Mandatory (Yes/No)	Description
key	str	No if used as a request parameter	Explanation: Object name to use when a 4XX error occurs. This parameter specifies the webpage to display when an error occurs. Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 7-136 RoutingRule

Parameter	Type	Mandatory (Yes/No)	Description
condition	Condition	No if used as a request parameter	Explanation: Conditions that must be met for the specified redirect to apply Value range: See Table 7-137 . Default value: None
redirect	Redirect	Yes if used as a request parameter	Explanation: Details about the redirection. For details, see Table 7-138 . Default value: None

Table 7-137 Condition

Parameter	Type	Mandatory (Yes/No)	Description
keyPrefixEquals	str	No if used as a request parameter	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 httpErrorReturnedEquals . Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
httpErrorCodeReturnedEquals	int	No if used as a request parameter	Explanation: HTTP error code for the redirection to take effect. If there is an error, and the 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. Default value: None

Table 7-138 Redirect

Parameter	Type	Mandatory (Yes/No)	Description
protocol	str	No if used as a request parameter	Explanation: Protocol used for redirection Value range: • http • https Default value: None
hostName	str	No if used as a request parameter	Explanation: Host name used for redirection Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
replaceKeyPrefix- With	str	No if used as a request parameter	Explanation: Object name prefix used in the redirection request Value range: The value must contain 1 to 1,024 characters. Default value: None
replaceKeyWith	str	No if used as a request parameter	Explanation: Object name used in the redirection request Restrictions: This parameter cannot be used together with replaceKeyPrefix- With . Value range: The value must contain 1 to 1,024 characters. Default value: None
httpRedirectCode	int	No if used as a request parameter	Explanation: HTTP status code in the response to the redirect request. Default value: None

Code Examples

This example returns the static website hosting configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"
```

```
# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the website configuration of the bucket.
    resp = obsClient.getBucketWebsite(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Website Succeeded')
        print('requestId:', resp.requestId)
        if resp.body.redirectAllRequestTo:
            print('redirectAllRequestTo.hostName:', resp.body.redirectAllRequestTo.hostName,
                ',redirectAllRequestTo.protocol:', resp.body.redirectAllRequestTo.protocol)
        if resp.body.indexDocument:
            print('indexDocument.suffix:', resp.body.indexDocument.suffix)
        if resp.body.errorDocument:
            print('errorDocument.key:', resp.body.errorDocument.key)
        if resp.body.routingRules:
            index = 1
            for rout in resp.body.routingRules:
                print('routingRule[', index, ':')
                index += 1
                print('condition.keyPrefixEquals:', rout.condition.keyPrefixEquals,
                    ',condition.httpErrorCodeReturnedEquals:', rout.condition.httpErrorCodeReturnedEquals)
                print('redirect.protocol:', rout.redirect.protocol, 'redirect.hostName:', rout.redirect.hostName,
                    ',redirect.replaceKeyPrefixWith:', rout.redirect.replaceKeyPrefixWith,
                    ',redirect.replaceKeyWith:', rout.redirect.replaceKeyWith, 'redirect.httpRedirectCode:',
                    rout.redirect.httpRedirectCode)
        else:
            print('Get Bucket Website Failed')
            print('requestId:', resp.requestId)
            print('errorCode:', resp.errorCode)
            print('errorMessage:', resp.errorMessage)
    except:
        print('Get Bucket Website Failed')
        print(traceback.format_exc())
```

7.26 Deleting Static Website Hosting for a Bucket (SDK for Python)

Function

You can host static website resources such as HTML web pages, flash files, as well as 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 configuration of the bucket.

Restrictions

- To delete the static website hosting configuration 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).

Method

ObsClient.deleteBucketWebsite(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandator y (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-139 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example deletes the static website hosting configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Delete the static website hosting configuration of the bucket.
    resp = obsClient.deleteBucketWebsite(bucketName)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Website Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Website Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Website Failed')
    print(traceback.format_exc())
```

7.27 Configuring Versioning for a Bucket (SDK for Python)

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.

This API configures the versioning status 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).

Method

`ObsClient.setBucketVersioning(bucketName, status, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
status	str	Yes	Explanation: Versioning status of the bucket Value range: Versioning status of the bucket. Possible values are: • Enabled • Suspended Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-140 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None

Parameter	Type	Description
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures versioning for bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Configure versioning for the bucket.
    resp = obsClient.setBucketVersioning(bucketName, status='Enabled')

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Versioning Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Versioning Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
```

```
print('Set Bucket Versioning Failed')  
print(traceback.format_exc())
```

7.28 Obtaining the Versioning Status of a Bucket (SDK for Python)

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.

This API obtains the versioning status of a bucket.

Restrictions

- To obtain 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).

Method

`ObsClient.getBucketVersioning(bucketName, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-141 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
str	Explanation: Versioning status of the bucket Value range: • Enabled • Suspended Default value: None

Code Examples

This example returns the versioning status of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the bucket's versioning status.
    resp = obsClient.getBucketVersioning(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Versioning Succeeded')
        print('requestId:', resp.requestId)
        print('status:', resp.body)
    else:
        print('Get Bucket Versioning Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Versioning Failed')
    print(traceback.format_exc())
```

7.29 Configuring CORS for a Bucket (SDK for Python)

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 general 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.

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).

Method

`ObsClient.setBucketCors(bucketName, corsRuleList, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
corsRuleList	list of CorsRule	Yes	Explanation: CORS rule list of the bucket. For details, see Table 7-142. Restrictions: A list can have a maximum of 100 CORS rules. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-142 CorsRule

Parameter	Type	Mandatory (Yes/No)	Description
id	str	No if used as a request parameter	Explanation: CORS rule ID Value range: The value must contain 1 to 255 characters. Default value: None
allowedMethod	list of str	Yes if used as a request parameter	Explanation: The allowed HTTP methods (types of operations on buckets and objects) for a cross-origin request. Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
allowedOrigin	list of str	Yes if used as a request parameter	Explanation: The origin from which the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com. Default value: None
allowedHeader	list of str	No if used as a request parameter	Explanation: 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
maxAgeSecond	int or str	No if used as a request parameter	Explanation: Time your client can cache the response for a cross-origin request Restrictions: Each CORS rule can contain at most one maxAgeSecond. Value range: An integer greater than or equal to 0, in seconds Default value: 100

Parameter	Type	Mandatory (Yes/No)	Description
exposeHeader	list of str	No if used as a request parameter	Explanation: 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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-143 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None

Parameter	Type	Description
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures CORS rules for bucket **examplebucket**.

```
from obs import ObsClient
from obs import CorsRule
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify CORS rules.
    cors1 = CorsRule(id='rule1', allowedMethod=['PUT', 'POST', 'GET', 'DELETE', 'HEAD'],
                    allowedOrigin=['obs.hostname', 'obs.hostname1'], allowedHeader=['obs-header-1'],
                    maxAgeSecond=60)
    cors2 = CorsRule(id='rule2', allowedMethod=['PUT', 'POST', 'GET'],
                    allowedOrigin=['obs.hostname', 'obs.hostname1'], allowedHeader=['header-1', 'header-2'],
                    maxAgeSecond=50, exposeHeader=['head1'])
    corsList = [cors1, cors2]
    bucketName = "examplebucket"
    # Configure CORS for the bucket.
    resp = obsClient.setBucketCors(bucketName, corsList)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
```

```
if resp.status < 300:
    print('Set Bucket Cors Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Set Bucket Cors Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Cors Failed')
    print(traceback.format_exc())
```

7.30 Obtaining the CORS Configuration of a Bucket (SDK for Python)

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 general 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).

Method

`ObsClient.getBucketCors(bucketName, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-144 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
list of CorsRule	Explanation: CORS rule list of the bucket. For details, see Table 7-145 .

Table 7-145 CorsRule

Parameter	Type	Mandatory (Yes/No)	Description
id	str	No if used as a request parameter	Explanation: CORS rule ID Value range: The value must contain 1 to 255 characters. Default value: None
allowedMethod	list of str	Yes if used as a request parameter	Explanation: The allowed HTTP methods (types of operations on buckets and objects) for a cross-origin request. Value range: The following HTTP methods are supported: • GET • PUT • HEAD • POST • DELETE Default value: None
allowedOrigin	list of str	Yes if used as a request parameter	Explanation: The origin from which the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
allowedHeader	list of str	No if used as a request parameter	Explanation: 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
maxAgeSecond	int or str	No if used as a request parameter	Explanation: Time your client can cache the response for a cross-origin request Restrictions: Each CORS rule can contain at most one maxAgeSecond. Value range: An integer greater than or equal to 0, in seconds Default value: 100
exposeHeader	list of str	No if used as a request parameter	Explanation: 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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Code Examples

This example returns the CORS configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Obtain the bucket's CORS configuration.
    resp = obsClient.getBucketCors(bucketName)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Cors Succeeded')
        print('requestId:', resp.requestId)
        index = 1
        for rule in resp.body:
            print('corsRule [' + str(index) + ']')
            print('id:', rule.id)
            print('allowedMethod', rule.allowedMethod)
            print('allowedOrigin', rule.allowedOrigin)
            print('allowedHeader', rule.allowedHeader)
            print('maxAgeSecond', rule.maxAgeSecond)
            print('exposeHeader', rule.exposeHeader)
            index += 1
    else:
        print('Get Bucket Cors Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Bucket Cors Failed')
    print(traceback.format_exc())
```

7.31 Deleting the CORS Configuration of a Bucket (SDK for Python)

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 general 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 configuration 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).

Method

`ObsClient.deleteBucketCors(bucketName, extensionHeaders)`

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-146 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example deletes CORS rules of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Delete CORS rules of the bucket.
    resp = obsClient.deleteBucketCors(bucketName)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Cors Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Cors Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Cors Failed')
    print(traceback.format_exc())
```

7.32 Configuring Tags for a Bucket (SDK for Python)

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 running data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags in SDRs.

This API adds tags to a bucket.

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 key values cannot contain commas (,), asterisks (*), vertical bars (|), slashes (/), less-than signs (<), greater-than signs (>), equal signs (=), backslashes (\), or ASCII codes (0x00 to 0x1F).
- To configure tags for a bucket, you must be the bucket owner or have the required permission (**obs:bucket:PutBucketTagging** in IAM or **PutBucketTagging** in a bucket policy).

Method

ObsClient.setBucketTagging(bucketName, tagInfo, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
tagInfo	TagInfo	Yes	Explanation: Bucket tag list. For details, see Table 7-147. Restrictions: - A bucket can have a maximum of 10 tags. Each tag can have only one pair of key values. - For the same bucket, tag keys must be unique, but tag values can be duplicated or left blank. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 7-147 TagInfo

Parameter	Type	Mandatory (Yes/No)	Description
tagSet	list of Tag	Yes if used as a request parameter	Explanation: Bucket tag list. For details, see Table 7-148 . Restrictions: - A bucket can have a maximum of 10 tags. Each tag can have only one pair of key values. - For the same bucket, tag keys must be unique, but tag values can be duplicated or left blank. Default value: None

Table 7-148 Tag

Parameter	Type	Mandatory (Yes/No)	Description
key	str	Yes if used as a request parameter	Explanation: Tag key Restrictions: - The tag key in the same bucket must be unique. - The value of the key can be self-defined or predefined by 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 value is case-sensitive. Default value: None
value	str	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. Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-149 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures tags for bucket **examplebucket**.

```
from obs import ObsClient
from obs import TagInfo
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```



```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    tagInfo = TagInfo()
    # Specify the label content.
    tagInfo.addTag('tag1', 'value1').addTag('tag2', 'value2')
    bucketName="examplebucket"
    # Configure tags for the bucket.
    resp = obsClient.setBucketTagging(bucketName, tagInfo)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Bucket Tagging Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Tagging Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Taggingg Failed')
    print(traceback.format_exc())
```

7.33 Obtaining Bucket Tags (SDK for Python)

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 running data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags in SDRs.

This API returns the tags of a bucket.

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).

Method

ObsClient.getBucketTagging(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-150 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

GetResult.body Type	Description
TagInfo	Explanation: Bucket tag configuration. For details, see Table 7-151 .

Table 7-151 TagInfo

Parameter	Type	Mandatory (Yes/No)	Description
tagSet	list of Tag	Yes if used as a request parameter	Explanation: Bucket tag list. For details, see Table 7-152. Restrictions: - A bucket can have a maximum of 10 tags. Each tag can have only one pair of key values. - For the same bucket, tag keys must be unique, but tag values can be duplicated or left blank. Default value: None

Table 7-152 Tag

Parameter	Type	Mandatory (Yes/No)	Description
key	str	Yes if used as a request parameter	Explanation: Tag key Restrictions: - The tag key in the same bucket must be unique. - The value of the key can be self-defined or predefined by 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 value is case-sensitive. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
value	str	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. Default value: None

Code Examples

This example returns tags of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain bucket tags.
    resp = obsClient.getBucketTagging(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Tagging Succeeded')
        print('requestId:', resp.requestId)
        index = 1
        for tag in resp.body.tagSet:
            print('tag [' + str(index) + ']')
```

```
        print('key:', tag.key)
        print('value:', tag.value)
        index += 1
    else:
        print('Get Bucket Tagging Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
    except:
        print('Get Bucket Tagging Failed')
        print(traceback.format_exc())
```

7.34 Deleting Bucket Tags (SDK for Python)

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 running data to a bucket, you can tag the bucket with the application name. In this manner, the costs on the application can be analyzed using tags in SDRs.

This API deletes the tags of a bucket.

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).

Method

ObsClient.deleteBucketTagging(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Responses

Type	Description
GetResult	Explanation: SDK common results

Table 7-153 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example deletes tags of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Delete bucket tags.
    resp = obsClient.deleteBucketTagging(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Tagging Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Tagging Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Tagging Failed')
    print(traceback.format_exc())
```

7.35 Obtaining the Custom Domain Name of a Bucket (SDK for Python)

Function

This API obtains the custom domain name of a bucket.

Restrictions

- To obtain the custom domain name of a bucket, you must be the bucket owner or have the required permission (**obs:bucket:GetBucketCustomDomainConfiguration** in IAM or **GetBucketCustomDomainConfiguration** in a bucket policy).

Method

ObsClient.getBucketCustomDomain(bucketName, extensionHeaders)

Request Parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 an IP address. – Cannot start or end with a hyphen (-) or period (.). – Cannot contain two consecutive periods (..), for example, my..bucket. – Cannot contain periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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: None Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Restrictions: None Value range: See User-defined Headers. Default value: None

Responses

Table 7-154 Responses

Type	Description
GetResult	Explanation: SDK common result objects

Table 7-155 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 7-156 Response result

GetResult.body Type	Description
GetBucketCustomDomainResponse	Explanation: Response result of obtaining the custom domain name of a bucket. For details, see GetBucketCustomDomainResponse .

Table 7-157 GetBucketCustomDomainResponse

Parameter	Type	Description
domains	list of Domain	Explanation: The bucket's custom domain name list. Value range: For details, see Domain .

Table 7-158 Domain

Parameter	Type	Description
name	str	Explanation: Name of the certificate bound to the custom domain name. Value range: The value is a string of 3 to 63 characters.
domain_name	str	Explanation: Custom domain name. Value range: The value is a string of 0 to 256 characters.

Parameter	Type	Description
create_time	str	Explanation: Time when a custom domain name was created. Value range: - The value must be a UTC time that complies with the ISO 8601 standard. Example: 2018-01-01T00:00:00.000Z - The value is a string of 24 characters.
certificate_id	str	Explanation: ID of the certificate bound to the custom domain name. Value range: The value is a string of 16 characters.
certificateType	str	Explanation: Type of the certificate bound to the custom domain name. Value range: server: server certificate server_sm: server SM certificate client: client CA certificate

Parameter	Type	Description
expiredTime	str	Explanation: If the certificate type is server or server_sm , the expiration time of the certificate bound to the custom domain name is returned. Value range: The value must be a UTC time that complies with the ISO 8601 standard. Example: 2018-01-01T00:00:00.000Z

Code Examples

This example obtains the custom domain name of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Obtain the custom domain name of the bucket.
    resp = obsClient.getBucketCustomDomain(bucketName)

    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Get Bucket CustomDomain Succeeded')
        print('requestId:', resp.requestId)
        for domin in resp.body.domains:
            print('domainName:', domin.domainName)
            print('createTime:', domin.createTime)
    else:
        print('Get Bucket CustomDomain Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
```

```
print('Get Bucket CustomDomain Failed')  
print(traceback.format_exc())
```

8 Object-Related APIs (SDK for Python)

8.1 Object Upload Overview (SDK for Python)

You can use this API to upload an object to a specified bucket. In OBS, objects are basic data units that you can operate. OBS Python SDK provides many APIs for uploading objects using various methods:

- **Text-based upload:** Character strings are used as the data source of an object.
- **Streaming upload:** Readable objects that contain the **read** attribute are used as the data source of an object.
- **File-based upload:** Local files are used as the data source of objects.
- **Multipart upload:** Large files can be uploaded in multiple parts.
- **Append upload:** You can append data to an object.
- **Resumable upload:** It 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.
- **Browser-based upload:** You can upload objects to a specified bucket in HTML form.

SDK supports the upload of objects whose size ranges from 0 KB to 5 GB. For streaming upload, append upload, and file-based upload, data to be uploaded in a batch 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 succeeds. 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 a bucket, its URL does not contain a directory level.

8.2 Uploading Objects - Text-Based (SDK for Python)

Function

This API uploads the text using strings as the data source to a specified 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).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, **multipart uploads** should be used.

Method

```
ObsClient.putContent(bucketName, objectKey, content, metadata, headers, progressCallback, autoClose,  
extensionHeaders)
```

Request Parameters

Table 8-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
content	str or readable object	No	Explanation: Content of the object to be uploaded Value range: A string or a readable object NOTE If content is a readable object that contains the read attribute, data is read from the readable object. Otherwise, the object content is a string. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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
headers	PutObject Header	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-2. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python). Upload progress callback only supports streaming, file-based, multipart, appendable, and resumable uploads.
autoClose	bool	No	Explanation: Whether or not to automatically close data streams after the upload is complete Value range: True: The data stream is automatically closed. False: The data stream is not automatically closed. Default value: True
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-2 PutObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Value range: Base64-encoded 128-bit MD5 value of the request body calculated according to RFC 1864 Example: n58IG6hfM7vqI4K0vnWpog== Default value: None
acl	str	No	Explanation: ACL that can be pre-defined during object creation Value range: For details about the available options, see HeadPermission. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
contentType	str	No	Explanation: Multipurpose Internet Mail Extensions (MIME) type of the object 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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: If you do not specify contentType 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 contentType.
contentLength	int	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, multipart uploads should be used. Default value: If this parameter is not specified, OBS SDK for Python automatically calculates the size of the object.
storageClass	str	No	Explanation: Storage class of the object to be uploaded Value range: See Table 8-4. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. - If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object to be uploaded Value range: See Table 8-5. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-3 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-4 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-5 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeId	str	No	Explanation: Account (domain) ID of the grantee Value range: See How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-6 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Responses

Table 8-7 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-8 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-9 GetResult.body

GetResult.body Type	Description
PutContentResponse	Explanation: Response to the request for uploading an object. For details, see Table 8-10 .

Table 8-10 PutContentResponse

Parameter	Type	Description
storageClass	str	Explanation: Storage class of the object to be uploaded Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-4. Default value: None
versionId	str	Explanation: Object version ID. If versioning is enabled for the bucket, the object version ID will be returned. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
etag	str	Explanation: ETag of an object, which is a base64-encoded 128-bit MD5 digest. ETag is the unique identifier of the object content. It can be used to determine whether the object content is changed. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, the object content is changed. The ETag reflects changes only to the contents of the object, not its metadata. An uploaded object or copied object 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
objectUrl	str	Explanation: Full path to the object Default value: None

Code Examples

This example uploads a text.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
```

```
bucketName = "examplebucket"
objectKey = "objectname"
# Specify the text to be uploaded.
content = 'Hello OBS'
# Upload the text.
resp = obsClient.putContent(bucketName, objectKey, content)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Put Content Succeeded')
    print('requestId:', resp.requestId)
    print('etag:', resp.body.etag)
else:
    print('Put Content Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

8.3 Uploading an Object - Streaming (SDK for Python)

Function

You can upload texts, images, videos, or any other types of files smaller than 5 GB.

Streaming upload uses readable objects that contain the **read** attribute as the data source and uploads data to a specified bucket in network streams or file streams.

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).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, **multipart uploads** should be used.

Method

```
ObsClient.putContent(bucketName, objectKey, content, metadata, headers, progressCallback, autoClose,
extensionHeaders)
```

Request Parameters

Table 8-11 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my.-bucket or my.bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
content	str or readable object	No	Explanation: Content of the object to be uploaded Value range: NOTICE - Streaming upload requires CONTENT to be a readable object that contains the READ attribute. - When uploading file streams, you must open files in RB or RB+ mode. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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
headers	PutObject Header	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-12. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python). Upload progress callback only supports streaming, file-based, multipart, appendable, and resumable uploads.
autoClose	bool	No	Explanation: Whether or not to automatically close data streams after the upload is complete Value range: True: The data stream is automatically closed. False: The data stream is not automatically closed. Default value: True
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-12 PutObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Value range: Base64-encoded 128-bit MD5 value of the request body calculated according to RFC 1864 Example: n58IG6hfM7vqI4K0vnWpog== Default value: None
acl	str	No	Explanation: ACL that can be pre-defined during object creation Value range: For details about the available options, see HeadPermission. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: If you do not specify contentType 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 contentType.

Parameter	Type	Mandatory (Yes/No)	Description
contentLength	int	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, multipart uploads should be used. Default value: If this parameter is not specified, OBS SDK for Python automatically calculates the size of the object.
storageClass	str	No	Explanation: Storage class of the object Value range: See Table 8-14. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. - If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object to be uploaded Value range: See Table 8-15 . Default value: None
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-13 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-14 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-15 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeld	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-16 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Responses

Table 8-17 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-18 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-19 GetResult.body

GetResult.body Type	Description
PutContentResponse	Explanation: Response to the request for uploading an object. For details, see Table 8-10 .

Table 8-20 PutContentResponse

Parameter	Type	Description
storageClass	str	Explanation: Storage class of the object Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-14. Default value: None
versionId	str	Explanation: Object version ID. If versioning is enabled for the bucket, the object version ID will be returned. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
etag	str	Explanation: ETag of an object, which is a base64-encoded 128-bit MD5 digest. ETag is the unique identifier of the object content. It can be used to determine whether the object content is changed. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, the object content is changed. The ETag reflects changes only to the contents of the object, not its metadata. An uploaded object or copied object 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
objectUrl	str	Explanation: Full path to the object Default value: None

Code Examples

This example uploads a network stream.

```
from obs import ObsClient
import os
import traceback
import sys
if sys.version_info.major == 2 or not sys.version > '3':
    import httplib
else:
    import http.client as httplib

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
```

```
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify a network stream.
    conn = httplib.HTTPConnection('www.a.com', 80)
    // Obtain the network stream.
    conn.request('GET', '/')
    # Read the network stream.
    content = conn.getresponse()
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Upload the network stream.
    resp = obsClient.putContent(bucketName, objectKey, content)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put Content Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Put Content Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

This example uploads a file stream.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Read a file stream.
    content = open('localfile', 'rb')
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Upload the file stream.
    resp = obsClient.putContent(bucketName, objectKey, content)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put Content Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Put Content Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

8.4 Uploading an Object - File-Based (SDK for Python)

Function

This API uploads local files to OBS over the Internet. These files can be texts, images, videos, or any other type of files.

NOTE

- OBS does not involve folders like in a file system. All elements stored in OBS buckets are objects. To create a folder in OBS is essentially to create an object whose size is 0 and whose name ends with a slash (/). You can perform download, delete, or other operations on such objects as you do on ordinary objects.
- If versioning is enabled, when uploading an object, OBS automatically allocates a unique version ID for the object. Objects with the same name are stored in OBS as objects with different version IDs. If versioning is not enabled, when uploading an object to a folder where there is already an object with the same name, the new object will overwrite the existing one.
- You can pass user-defined headers in **extensionHeaders** in a dictionary. For details, see [User-defined Headers \(SDK for Python\)](#).

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).
- The object size in a single upload ranges from 0 to 5 GB.
- To upload files larger than 5 GB, [multipart uploads](#) should be used.

Method

```
ObsClient.putFile(bucketName, objectKey, file_path, metadata, headers, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-21 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
file_path	str	Yes	Explanation: Full path of the file or folder to be uploaded, for example, aa/bb.txt or aa/. Default value: None NOTE If file_path is a folder, contentLength, md5, and contentType in headers cannot take effect.

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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
headers	PutObject Header	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-22. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Restrictions: The progress of a folder upload cannot be obtained. Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python). Upload progress callback only supports streaming, file-based, multipart, appendable, and resumable uploads.
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-22 PutObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Value range: Base64-encoded 128-bit MD5 value of the request body calculated according to RFC 1864 Example: n58IG6hfM7vql4K0vnWpog== Default value: None
acl	str	No	Explanation: ACL that can be pre-defined during object creation Value range: For details about the available options, see HeadPermission. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: If you do not specify contentType 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 contentType.

Parameter	Type	Mandatory (Yes/No)	Description
contentLength	int	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, multipart uploads should be used. Default value: If this parameter is not specified, OBS SDK for Python automatically calculates the size of the object.
storageClass	str	No	Explanation: Storage class of the object Value range: See Table 8-24. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. - If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object to be uploaded Value range: See Table 8-25 . Default value: None
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-23 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-24 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-25 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeld	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-26 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Responses

Table 8-27 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-28 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-29 GetResult.body

GetResult.body Type	Description
PutContentResponse	Explanation: Response to the request for uploading an object. For details, see Table 8-30 . Default value: None

 NOTE

If **file_path** is a folder, the returned result is a **GetResult** list.

Table 8-30 PutContentResponse

Parameter	Type	Description
storageClass	str	Explanation: Storage class of the object Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-24. Default value: None

Parameter	Type	Description
versionId	str	Explanation: Object version ID. If versioning is enabled for the bucket, the object version ID will be returned. Value range: The value must contain 32 characters. Default value: None
etag	str	Explanation: ETag of an object, which is a base64-encoded 128-bit MD5 digest. ETag is the unique identifier of the object content. It can be used to determine whether the object content is changed. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, the object content is changed. The ETag reflects changes only to the contents of the object, not its metadata. An uploaded object or copied object 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
objectUrl	str	Explanation: Full path to the object Default value: None

Code Example 1: Uploading a Single File

This example uploads a single file.

```
from obs import ObsClient
from obs import PutObjectHeader
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
```

```
SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional headers of the request for uploading an object.
    headers = PutObjectHeader()
    # (Optional) Specify the MIME type of the object.
    headers.contentType = 'text/plain'
    bucketName = "examplebucket"
    # Specify an object name (the name displayed after the file is uploaded to the bucket).
    objectKey = "objectname"
    # Specify the full path of the file to be uploaded, for example, aa/bb.txt.
    file_path = 'localfile'
    # Specify the custom metadata of the object.
    metadata = {'meta1': 'value1', 'meta2': 'value2'}
    # Perform the file-based upload.
    resp = obsClient.putFile(bucketName, objectKey, file_path, metadata, headers)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put File Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
        print('versionId:', resp.body.versionId)
        print('storageClass:', resp.body.storageClass)
    else:
        print('Put File Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put File Failed')
    print(traceback.format_exc())
```

Code Example 2: Uploading a Folder

This example uploads all files in a folder. The **putFile** method does not support concurrent uploads. If you need to upload all files in a folder concurrently for better performance, see Code Example 3.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
```

```
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

def out_put_res(resp, objectKey=None):
    if isinstance(resp, list):
        for res in resp:
            out_put_res(res)
    elif isinstance(resp, tuple) and isinstance(resp[1], list):
        out_put_res(resp[1])
    elif isinstance(resp, tuple):
        if resp[1].status < 300:
            print(f'Put File Succeeded, objectkey: {resp[0]}')
        else:
            print(f'Put File Failed, objectkey: {resp[0]}')
            print('requestId:', resp[1].requestId)
            print('errorCode:', resp[1].errorCode)
            print('errorMessage:', resp[1].errorMessage)
    else:
        if resp.status < 300:
            print(f'Put File Succeeded, objectkey: {objectKey}')
        else:
            print(f'Put File Failed, objectkey: {objectKey}')
            print('requestId:', resp.requestId)
            print('errorCode:', resp.errorCode)
            print('errorMessage:', resp.errorMessage)

try:
    bucketName = "examplebucket"
    # Specify a name for the uploaded folder. All files in the local folder are uploaded to this folder. Its name
    # cannot end with a slash (/).
    objectKey = "folder"
    # Specify the full path of the folder to be uploaded, for example, aa/.
    folder_path = 'localfolder/'
    # Upload the folder.
    resp = obsClient.putFile(bucketName, objectKey, folder_path)
    # resp is a list of upload results of each file in the folder.
    out_put_res(resp, objectKey)
except:
    print('Put File Failed')
    print(traceback.format_exc())
```

Code Example 3: Uploading the Files in a Folder Concurrently

This example uploads all files in a folder concurrently.

```
from obs import ObsClient
from concurrent.futures import ThreadPoolExecutor, as_completed
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
# SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
folder_path = 'localfolder/'
bucketName = 'examplebucket'
# Specify a name for the uploaded folder. All files in the local folder are uploaded to this folder. Its name
# must end with a slash (/). If you want to upload files to the root directory, enter an empty string for the
# prefix, that is, prefix = "".
```

```
prefix = 'testobs/'
ThreadNum = 20
g = os.walk(folder_path)
# Create a thread pool for upload.
pool = ThreadPoolExecutor(ThreadNum)
all_task = []
for path, dir_list, file_list in g:
    for file_name in file_list:
        srcKey = os.path.join(path, file_name)
        obsObjectKey = prefix + srcKey.split(folder_path)[1].replace('\\', '/')
        exists = False
        try:
            # (Optional) Check whether the file already exists on OBS based on the object name.
            # resp = obsClient.headObject(bucketName, obsObjectKey)
            # if resp.status < 300:
            #     exists = True
            # elif resp.status == 404:
            #     exists = False
            # else:
            #     print('Error happened, reupload it.')
            if not exists:
                print("File %s not exists in obs, upload it", srcKey)
                all_task.append(pool.submit(obsClient.putFile, bucketName, obsObjectKey, srcKey))
            # You are advised to use obsClient.uploadFile to upload large files. For details about the
            parameters, see the section about the API for resumable upload.
            # partSize = 9 * 1024 * 1024
            # taskNum = 10
            # enableCheckpoint = True
            # all_task.append(pool.submit(obsClient.uploadFile, bucketName, obsObjectKey, srcKey, partSize,
            taskNum, enableCheckpoint))
        except:
            print(traceback.format_exc())
for future in as_completed(all_task):
    put_resp = future.result()
    if put_resp.status < 300:
        print(f'Put File Succeeded, objectUrl: {put_resp.body.objectUrl}')
    else:
        print('Put File Failed')
        print('requestId:', put_resp.requestId)
        print('errorCode:', put_resp.errorCode)
        print('errorMessage:', put_resp.errorMessage)
```

8.5 Uploading an Object - Append (SDK for Python)

Function

This API uploads a file or folder to an existing OBS bucket. These files can be texts, images, videos, or any other type of files.

The **AppendObject** operation adds data to the end of an object in a specified bucket. If there is no object with the same key values in the bucket, a new object is created.

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).
- Uploaded objects are stored in buckets. To upload an object to a bucket, you must have the write permission for the bucket. The name of each object in a bucket must be unique.
- The latest modification time of the object is updated each time an upload is appended.

- If you use SSE-C encryption, the encryption header you configure, such as **x-obs-server-side-encryption**, must be carried in each append upload.
- If you use SSE-KMS encryption, the encryption header you configure, such as **x-obs-server-side-encryption**, only needs to be carried when the object is uploaded for the first time and no object with the same name exists in the bucket.
- The size of each append upload cannot exceed 5 GB.
- The maximum number of append writes for each appendable object is 10,000.
- If the storage class is , this API cannot be called.

Method

```
ObsClient.appendObject(bucketName, objectKey, content, metadata, headers, progressCallback, autoClose, extensionHeaders)
```

Request Parameters

Table 8-31 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
content	AppendObjectContent	Yes	Explanation: Content to be appended Value range: See Table 8-32. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	No	Explanation: Custom metadata to be appended. 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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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

Parameter	Type	Mandatory (Yes/No)	Description
headers	AppendObjectHeader	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Restrictions: This parameter is valid only for the first append upload. Value range: See Table 8-33. Default value: None
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python).
autoClose	bool	No	Explanation: Whether or not to automatically close data streams after the upload is complete Value range: True: The data stream is automatically closed. False: The data stream is not automatically closed. Default value: True

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-32 AppendObjectContent

Parameter	Type	Mandatory (Yes/No)	Description
content	str or readable object	No	Explanation: Content to be appended Value range: - A character string of object content - Readable object - Path of the file to be uploaded (isFile must be set to True.) NOTE If content is a readable object that contains the read attribute, data can be read from content . Otherwise, the object content is a character string. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
position	int or str	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 nextPosition returned in the response when the first upload is successful. Value range: An integer greater than or equal to 0, in bytes Default value: 0
offset	int or str	No	Explanation: Offset, in bytes. This parameter is required if the content for an append upload is a local file. Value range: An integer greater than or equal to 0, in bytes Default value: 0
isFile	bool	No	Explanation: Whether content indicates the file path. Value range: True: content indicates the file path. False: content does not indicate the file path. Default value: False

Table 8-33 AppendObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the content to be appended. It is used for the OBS server to verify data integrity. Value range: Base64-encoded 128-bit MD5 value of the request body calculated according to RFC 1864 Example: n58lG6hfM7vql4K0vnWpog== Default value: None
acl	str	No	Explanation: Pre-defined ACL, which can be specified in the append upload request. 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: For details about the available options, see HeadPermission. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. 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 for objects in the root directory of a bucket. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
contentType	str	No	Explanation: MIME type of the object specified in the first append upload. 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)? (Python SDK) 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. Default value: If you do not specify contentType 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 contentType.
contentLength	int	No	Explanation: Length of the content to be appended Restrictions: • The object size in a single upload ranges from 0 to 5 GB. • To upload files larger than 5 GB, multipart uploads should be used. Default value: If this parameter is not specified, OBS SDK for Python automatically calculates the size of the object.

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	No	Explanation: Storage class of the object that can be specified in the append upload request 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: • If the storage class is Standard, leave this parameter blank. • For details about the available storage classes, see Table 8-35. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. • If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionGrants	list of ExtensionGrant	No	Explanation: List of extension permissions that can be specified in the append upload request 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 8-36. Default value: The value specified in the first append upload request
expires	int	No	Explanation: Lifecycle (starting from the last modification time of the object) that can be specified in the append upload request. Once the object expires, it is automatically deleted. 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. - This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-34 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-35 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.

Parameter	Type	Description
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-36 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeId	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-37 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.

Constant	Description
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Responses

Table 8-38 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-39 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-40 GetResult.body

GetResult.body Type	Description
AppendObjectResponse	Explanation: Response to the append upload request. For details, see Table 8-41 .

Table 8-41 AppendObjectResponse

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-35. Default value: None
etag	str	Explanation: ETag of an object, which is a base64-encoded 128-bit MD5 digest. ETag is the unique identifier of the object content. It can be used to determine whether the object content is changed. For example, if the ETag is A when an object is uploaded and is B when the object is downloaded, the object content is changed. The ETag reflects changes only to the contents of the object, not its metadata. An uploaded object or copied object 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
nextPosition	int	Explanation: Start position for next appending Value range: An integer greater than or equal to 0, in bytes Default value: None
objectUrl	str	Explanation: Full path to the object Default value: None

Code Examples

This example appends content to an object.

```
from obs import ObsClient
from obs import AppendObjectContent
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the message body of the request for an append upload.
    content = AppendObjectContent()
    # Specify the content to be appended.
    content.content = 'Hello OBS'
    # Specify the starting position (byte 0 in this example) the content is appended to.
    content.position = 0
    # If you upload an object for the first time using the append upload, an error will be reported (status
    # code 409) if an ordinary object with the same name already exists.
    bucketName = "examplebucket"
    # Specify the name of the object to append content to.
    objectKey = "objectname"
    # Append content to the object.
    resp = obsClient.appendObject(bucketName, objectKey, content)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Append Object Succeeded')
        print('requestId:', resp.requestId)
        print('nextPosition:', resp.body.nextPosition)
    else:
        print('Append Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Append Object Failed')
    print(traceback.format_exc())
```

 NOTE

- Objects uploaded using **ObsClient.putObject**, referred to as common 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 common 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 the append upload is the ETag for the appended content, rather than that of the whole object.
- Data appended each time can be up to 5 GB, and a maximum of 10,000 uploads can be appended for an object.
- After an append upload is successful, you can obtain the location for the next append upload by using **body.nextPosition** in the returned results or call **ObsClient.getObjectMetadata**.

8.6 Uploading an Object - Resumable (SDK for Python)

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. This API splits the file into multiple parts and uploads them individually. The upload result of each part is recorded in a checkpoint file in real time. A success message is returned only when all parts are uploaded. If any parts fail, an error message is returned telling you to call the API again to upload the failed parts. Since the checkpoint file contains the progress of each part, it saves you uploading all parts again in the event of an error.

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).
- The file uploaded by the resumable upload API must exceed 100 KB.
- To obtain the progress of the last upload, you must enable resumable upload when you use this API.

Method

```
ObsClient.uploadFile(bucketName, objectKey, uploadFile, partSize, taskNum, enableCheckpoint,
checkpointFile, checksum, metadata, progressCallback, headers, extensionHeaders)
```

Request Parameters

Table 8-42 List of request parameters

Parameter	Type	Description	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Description	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
uploadFile	str	Yes	Explanation: Complete path of the local file to be uploaded, for example, aa/bb.txt . Default value: None
partSize	int	No	Explanation: Part size Value range: The value ranges from 100 KB to 5 GB, in bytes. Default value: 9 MB
taskNum	int	No	Explanation: Maximum number of parts that can be uploaded concurrently Value range: 1~10000 Default value: 1, indicating concurrent uploads are not used.

Parameter	Type	Description	Description
enableCheckpoint	bool	No	Explanation: Whether to enable the resumable upload mode Value range: True: The resumable upload mode is enabled. False: The resumable upload mode is disabled. Default value: False
checkpointFile	str	No	Explanation: Address of a file generated for recording the progress of a resumable upload. The file contains the information about parts and the upload progress. Restrictions: This parameter is valid only for resumable uploads. Default value: If this parameter is left blank, the progress file will be in the same directory as the local file to be uploaded.
checkSum	bool	No	Explanation: Whether to verify the file to upload. If this parameter is enabled, before each task restarts, the system verifies whether the file to upload is the one used during task initialization. Value range: True: The file to upload is verified. False: The file to upload is not verified. Default value: False

Parameter	Type	Description	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python).

Parameter	Type	Description	Description
headers	UploadFileHeader	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-43 . Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-43 UploadFileHeader

Parameter	Type	Description	Description
acl	str	No	Explanation: ACL that can be pre-defined during object creation Value range: For details about the available options, see HeadPermission . Default value: None

Parameter	Type	Description	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: If you do not specify contentType 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 contentType.

Parameter	Type	Description	Description
storageClass	str	No	Explanation: Storage class of the object Value range: For details about the available storage classes, see Table 8-45. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. • If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object to be uploaded Value range: See Table 8-46. Default value: None

Parameter	Type	Description	Description
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-44 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-45 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-46 ExtensionGrant

Parameter	Type	Description	Description
granteeId	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-47 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Responses

Table 8-48 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-49 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-50 GetResult.body

GetResult.body Type	Description
CompleteMultipartUploadResponse	Explanation: Response to the part assembling. For details, see Table 8-51 .

Table 8-51 CompleteMultipartUploadResponse

Parameter	Type	Description
etag	str	Explanation: The ETag that uniquely identifies the object after its parts were assembled, calculated based on the ETag of each part. 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

Parameter	Type	Description
bucket	str	Explanation: Bucket in which parts are assembled 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
key	str	Explanation: Object name obtained after part assembling. 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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
location	str	Explanation: URL of the generated object after part assembling Default value: None
versionId	str	Explanation: Version ID of the object obtained after part assembling Value range: The value must contain 32 characters. Default value: None
objectUrl	str	Explanation: Full path to the obtained object after part assembling Default value: None

Code Examples

This example uploads object **localfile** to bucket **examplebucket** using resumable upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

try:
    bucketName = "examplebucket"
    # Specify an object name (the name displayed after the file is uploaded to the bucket).
    objectKey = "objectname"
    # Specify the path of the file to be uploaded.
    uploadFile = 'localfile'
    # Specify the number of parts that can be concurrently uploaded.
    taskNum = 5
    # Specify the part size, in bytes. 10 MB is used as an example.
```

```
partSize = 10 * 1024 * 1024
# Enable the resumable upload by setting enableCheckpoint to True.
enableCheckpoint = True
# Upload the object using resumable upload.
resp = obsClient.uploadFile(bucketName, objectKey, uploadFile, partSize, taskNum, enableCheckpoint,
encoding_type='url')

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Upload File Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Upload File Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Upload File Failed')
    print(traceback.format_exc())
```

8.7 Obtaining the Upload Progress (SDK for Python)

You can query the upload progress when uploading an object in streaming, file-based, multipart, appendable, or resumable mode.

This example configures a callback function to obtain the object upload progress.

Sample code is as follows:

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

# Obtain the upload progress.
def callback(transferredAmount, totalAmount, totalSeconds):
    # Obtain the average upload rate (KB/s).
    print(transferredAmount * 1.0 / totalSeconds / 1024)
    # Obtain the upload progress in percentage.
    print(transferredAmount * 100.0 / totalAmount)

try:
    bucketName = "examplebucket"
    # Specify an object name (the name displayed after the file is uploaded to the bucket).
    objectKey = "objectname"
    # Specify the full path of the file to be uploaded, for example, aa/bb.txt.
    file_path = 'localfile'
    # Perform the file-based upload.
    resp = obsClient.putFile(bucketName, objectKey, file_path, progressCallback=callback)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put File Succeeded')
```

```
print('requestId:', resp.requestId)
print('etag:', resp.body.etag)
print('versionId:', resp.body.versionId)
print('storageClass:', resp.body.storageClass)
else:
    print('Put File Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Put File Failed')
    print(traceback.format_exc())
```

8.8 Uploading an Object - Browser-Based (SDK for Python)

Function

This API uploads an object up to 5 GB to a specified bucket in HTML form. You can call **ObsClient.createPostSignature** to generate parameters for requesting a browser-based upload.

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

NOTE

There are two request parameters generated for authentication:

- **policy**, which corresponds to the **policy** parameter in the form.
- **signature**, which corresponds to the **signature** parameter in the form.

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).
- Values of **policy** and **signature** in the HTML form are obtained from the returned results of **ObsClient.createPostSignature**.

Method

```
ObsClient.createPostSignature(bucketName, objectKey, expires, formParams)
```

Request Parameters

Table 8-52 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	No	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	No	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
expires	int	No	Explanation: Expiration time of authentication for a browser-based upload Value range: A positive integer, in seconds Default value: 300
formParams	dict	No	Explanation: Parameters of browser-based uploads, not including key, policy, and signature. Value range: (When you use the following parameters, you must add the x-obs prefix to them. Taking acl as an example, it should be configured as formParams['x-obs-acl']='public-read'. To obtain the values of these parameters, see the response header descriptions in APIs.) • acl • cache-control • content-type • content-disposition • content-encoding • expires Default value: None

Responses

Table 8-53 List of returned results

Parameter	Type	Description
originPolicy	str	Explanation: Value of Policy that is not encoded by Base64. This parameter can only be used for verification. For example: {"expiration":"2023-09-12T12:52:59Z","conditions":[{"content-type":"text/plain"}, {"bucket":"examplebucket"}, {"key":"example/objectname"}],}" Default value: None
policy	str	Explanation: Value of Policy that is encoded by Base64. For example: eyJleHBpcmF0aW9uljoiMjAyMy0wOS0xMlQxMjo1Mjo1OVoiLCJjb25kaXRpb25zljpbeyJjb250ZW50LXR5cGUiOiJ0ZXh0L3BsYWluIn0seyJidWNrZXQiOiJleGFtcGxlYnVja2V0In0seyJrZXkiOiJleGFtcGxlL29iamVjdG5hbWUifSxdfQ== Default value: None
signature	str	Explanation: signature in the form For example: g0jQr4v9VWd1Q2FOFDG6LGfV9Cw= Default value: None

Code Examples

This example generates authentication parameters **policy** and **signature** for a browser-based upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Configure the validity period (in seconds) for a browser-based upload request. 3600 is used as an
    example.
    expires = 3600
    # Specify parameters for a browser-based upload except key, policy, and signature. In this example, x-
    obs-acl is set to private and content-type is set to text/plain.
    formParams = {'x-obs-acl': 'private', 'content-type': 'text/plain'}
    # Create parameters for a browser-based upload.
    resp = obsClient.createPostSignature(bucketName, objectKey, expires, formParams)

    print('originPolicy:', resp.originPolicy)
    print('policy:', resp.policy)
    print('signature:', resp.signature)
except:
    print(traceback.format_exc())
```

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">
    <p>
      Object key
    </p>
    <!-- Object name -->
    <input type="text" name="key" value="objectname" />
    <p>
      ACL
    </p>
    <!-- Object ACL -->
    <input type="text" name="x-obs-acl" value="private" />
    <p>
      Content-Type
    </p>
    <!-- Object MIME type -->
    <input type="text" name="content-type" value="text/plain" />
    <p>
      <!-- Base64-encoded policy -->
      <input type="hidden" name="policy" value="*** Provide your policy ***" />
      <!-- AK -->
      <input type="hidden" name="AccessKeyId" value="*** Provide your access key ***" />
      <!-- Signature string information -->
      <input type="hidden" name="signature" value="*** Provide your signature ***" />
      <input name="file" type="file" />
      <input name="submit" value="Upload" type="submit" />
    </p>
  </form>
</body>
</html>
```

 NOTE

- Values of **policy** and **signature** in the HTML form are obtained from the returned results of **ObsClient.createPostSignature**.

8.9 Object Download Overview (SDK for Python)

OBS Python SDK provides abundant APIs for downloading objects using various methods:

- [Binary Download](#)
- [Streaming Download](#)
- [File-Based Download](#)
- [Partial Download](#)
- [Resumable Download](#)

8.10 Downloading an Object - Binary (SDK for Python)

Function

This API downloads a file using a stream from OBS to your local computer. If **loadStreamInMemory** is set to **True**, **downloadpath** will be invalid. The binary stream of the file will be downloaded to the 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).
- Objects in the storage class can be downloaded only when they are restored.

Method

```
ObsClient.getObject(bucketName, objectKey, downloadPath, getObjectRequest, headers,  
loadStreamInMemory, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-54 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
downloadPath	str	No	Explanation: The download path with the file name contained, for example, aa/bb.txt. Default value: None
getObjectRequest	GetObject Request	No	Explanation: Additional parameters of an object download request Value range: See Table 8-55. Default value: None
headers	GetObject Header	No	Explanation: Headers in the request used for obtaining the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-56. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
loadStreamInMemory	bool	No	Explanation: Whether to load the data stream of the object to memory Value range: True: The obtained data stream is loaded to memory with downloadPath ignored. False: The obtained data stream is not loaded to memory. Default value: False
progressCallback	callable	No	Explanation: Callback function for obtaining the download progress Default value: None NOTE This function contains the following parameters in sequence: number of downloaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Downloading an Object - Obtaining the Download Progress (SDK for Python).
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-55 GetObjectRequest

Parameter	Type	Mandatory (Yes/No)	Description
cache_control	str	No	Explanation: Cache-Control is rewritten in the response. Default value: None
content_disposition	str	No	Explanation: Content-Disposition is rewritten in the response. Default value: None
content_encoding	str	No	Explanation: Content-Encoding is rewritten in the response. Default value: None
content_language	str	No	Explanation: Content-Language is rewritten in the response. Default value: None
content_type	str	No	Explanation: Content-Type is rewritten in the response. Default value: None
expires	str	No	Explanation: Expires is rewritten in the response. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is obtained.

Table 8-56 GetObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
range	str	No	Explanation: Download range. For example, 0-999 indicates the download range is from byte 1 to byte 1,000. Value range: Value range: 0 to the object length minus 1. Format: <i>x-y</i>, indicating the range is from byte <i>x+1</i> to byte <i>y+1</i> Restrictions: The upper limit 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_modified_since	str or DateTime	No	Explanation: The object is returned if it has been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-57 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_unmodified_since	str or DateTime	No	Explanation: The object is returned if it has not been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-57 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: Each origin can contain only one wildcard character (*). Default value: None
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Table 8-57 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

Responses

Table 8-58 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-59 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-60 GetResult.body

GetResult.body Type	Description
ObjectStream	Explanation: Response to the request for downloading an object

Table 8-61 ObjectStream

Parameter	Type	Description
response	object	Explanation: If loadStreamInMemory is set to False and downloadPath is left blank in the request, this parameter is returned and indicates a readable stream. You can read object content from it. Default value: None
buffer	object	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the data stream in the memory. Default value: None
size	int	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the size of the data stream. Value range: An integer greater than or equal to 0, in bytes Default value: None
url	str	Explanation: If loadStreamInMemory is set to False and downloadPath is not left blank, this parameter is returned and indicates the download path. Default value: None
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. Default value: False

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-62. Default value: None
accessContorlAllo- wOrigin	str	Explanation: If Origin in the request meets the CORS rules of the bucket, AllowedOrigin in the CORS rules is returned. AllowedOrigin indicates the origin from which the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com. Default value: None
accessContorlAllow- Headers	str	Explanation: If RequestHeader in the request meets the CORS rules of the bucket, AllowedHeader 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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlAllowMethods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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
accessControlExposeHeaders	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlMaxAge	int	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 only one MaxAgeSeconds . Value range: An integer greater than or equal to 0, in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
cacheControl	str	Explanation: Cache-Control header in the response Default value: None
contentDisposition	str	Explanation: Content-Disposition header in the response Default value: None
contentEncoding	str	Explanation: Content-Encoding header in the response Default value: None
contentType	str	Explanation: Content-Type header in the response Default value: None

Parameter	Type	Description
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
expires	str	Explanation: Expires header in the response Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT . Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: This parameter is only available for objects. Default value: None

Parameter	Type	Description
expiration	str	Explanation: Expiration details Example: "expiry-date=\\"Mon, 11 Sep 2023 00:00:00 GMT\\"" Default value: None
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation: anotherPage.html The request is redirected to an external URL http://www.example.com/: 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 for objects in the root directory of a bucket. Default value: None

Table 8-62 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example downloads an object in binary mode.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    objectKey="objectname"
    # If loadStreamInMemory is set to True, downloadpath will be invalid. The binary stream of the file
    # will be downloaded to the memory.
    #Download an object in a binary stream.
    resp = obsClient.getObject(bucketName=bucketName,objectKey=objectKey, loadStreamInMemory=True)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp.requestId)
        # Obtain the object content.
        print('buffer:', resp.body.buffer)
        print('size:', resp.body.size)
    else:
        print('Get Object Failed')
        print('requestId:', resp.requestId)
```

```
print('errorCode:', resp.errorCode)
print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

NOTE

In a binary download, if **loadStreamInMemory** is set to **True**, the object content is contained in the **body.buffer** parameter in the returned result.

8.11 Downloading an Object - Streaming (SDK for Python)

Function

This API downloads a file using a stream from OBS to your local computer. If the object to be downloaded is too large or the download is expected to take a long time, you can use streams to download the object.

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).
- Objects in the storage class can be downloaded only when they are restored.

Method

```
ObsClient.getObject(bucketName, objectKey, downloadPath, getObjectRequest, headers,
loadStreamInMemory, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-63 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
downloadPath	str	No	Explanation: The download path with the file name contained, for example, aa/bb.txt. Default value: None
getObjectRequest	GetObject Request	No	Explanation: Additional parameters of an object download request Value range: See Table 8-64. Default value: None
headers	GetObject Header	No	Explanation: Headers in the request used for obtaining the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-65. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
loadStreamInMemory	bool	No	Explanation: Whether to load the data stream of the object to memory Value range: True: The obtained data stream is loaded to memory with downloadPath ignored. False: The obtained data stream is not loaded to memory. Default value: False
progressCallback	callable	No	Explanation: Callback function for obtaining the download progress Default value: None NOTE This function contains the following parameters in sequence: number of downloaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Downloading an Object - Obtaining the Download Progress (SDK for Python).
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-64 GetObjectRequest

Parameter	Type	Mandatory (Yes/No)	Description
cache_control	str	No	Explanation: Cache-Control is rewritten in the response. Default value: None
content_disposition	str	No	Explanation: Content-Disposition is rewritten in the response. Default value: None
content_encoding	str	No	Explanation: Content-Encoding is rewritten in the response. Default value: None
content_language	str	No	Explanation: Content-Language is rewritten in the response. Default value: None
content_type	str	No	Explanation: Content-Type is rewritten in the response. Default value: None
expires	str	No	Explanation: Expires is rewritten in the response. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is obtained.

Table 8-65 GetObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
range	str	No	Explanation: Download range. For example, 0-999 indicates the download range is from byte 1 to byte 1,000. Value range: Value range: 0 to the object length minus 1. Format: <i>x-y</i>, indicating the range is from byte <i>x</i>+1 to byte <i>y</i>+1 Restrictions: The upper limit 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_modified_since	str or DateTime	No	Explanation: The object is returned if it has been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-66 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_unmodified_since	str or DateTime	No	Explanation: The object is returned if it has not been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-66 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: Each origin can contain only one wildcard character (*). Default value: None
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Table 8-66 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

List of returned results

Table 8-67 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-68 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-69 GetResult.body

GetResult.body Type	Description
ObjectStream	Explanation: Response to the request for downloading an object

Table 8-70 ObjectStream

Parameter	Type	Description
response	object	Explanation: If loadStreamInMemory is set to False and downloadPath is left blank in the request, this parameter is returned and indicates a readable stream. You can read object content from it. Default value: None
buffer	object	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the data stream in the memory. Default value: None
size	int	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the size of the data stream. Value range: An integer greater than or equal to 0, in bytes Default value: None
url	str	Explanation: If loadStreamInMemory is set to False and downloadPath is not left blank, this parameter is returned and indicates the download path. Default value: None
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. Default value: False

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-71. Default value: None
accessContorlAllo- wOrigin	str	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 the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com. Default value: None
accessContorlAllow- Headers	str	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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlAllow-Methods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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
accessControlExpose-Headers	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlMaxAge	int	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 only one MaxAgeSeconds. Value range: An integer greater than or equal to 0, in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
cacheControl	str	Explanation: Cache-Control header in the response Default value: None
contentDisposition	str	Explanation: Content-Disposition header in the response Default value: None
contentEncoding	str	Explanation: Content-Encoding header in the response Default value: None
contentType	str	Explanation: Content-Type header in the response Default value: None

Parameter	Type	Description
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
expires	str	Explanation: Expires header in the response Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT. Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: Default value: None

Parameter	Type	Description
expiration	str	Explanation: Expiration details. Example: " expiry-date= \`Mon, 11 Sep 2023 00:00:00 GMT\`" Default value: None
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation: anotherPage.html The request is redirected to an external URL http://www.example.com/ : 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 for objects in the root directory of a bucket. Default value: None

Table 8-71 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example downloads an object using streaming.

```
from obs import GetObjectRequest
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional parameter of an object download request.
    getObjectRequest = GetObjectRequest()
    # Rewrite the Content-Type header in the response.
    getObjectRequest.content_type = 'text/plain'
    bucketName="examplebucket"
    objectKey="objectname"
    # Download the object using streaming.
    resp = obsClient.getObject(bucketName=bucketName,objectKey=objectKey,
    getObjectRequest=getObjectRequest, loadStreamInMemory=False)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp.requestId)
        # Read the object content.
        while True:
```

```
        chunk = resp.body.response.read(65536)
        if not chunk:
            break
        print(chunk)
    resp.body.response.close()
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

8.12 Downloading an Object - File-Based (SDK for Python)

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).
- Objects in the storage class can be downloaded only when they are restored.

Method

```
ObsClient.getObject(bucketName, objectKey, downloadPath, getObjectRequest, headers,
loadStreamInMemory, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-72 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
downloadPath	str	No	Explanation: The download path with the file name contained, for example, aa/bb.txt. Default value: None
getObjectRequest	GetObjectRequest	No	Explanation: Additional parameters of an object download request Value range: See Table 8-73.
headers	GetObjectHeader	No	Explanation: Headers in the request used for obtaining the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-74.

Parameter	Type	Mandatory (Yes/No)	Description
loadStreamInMemory	bool	No	Explanation: Whether to load the data stream of the object to memory Value range: True: The obtained data stream is loaded to memory with downloadPath ignored. False: The obtained data stream is not loaded to memory. Default value: False
progressCallback	callable	No	Explanation: Callback function for obtaining the download progress Default value: None NOTE This function contains the following parameters in sequence: number of downloaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Downloading an Object - Obtaining the Download Progress (SDK for Python).
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-73 GetObjectRequest

Parameter	Type	Mandatory (Yes/No)	Description
cache_control	str	No	Explanation: Cache-Control is rewritten in the response. Default value: None
content_disposition	str	No	Explanation: Content-Disposition is rewritten in the response. Default value: None
content_encoding	str	No	Explanation: Content-Encoding is rewritten in the response. Default value: None
content_language	str	No	Explanation: Content-Language is rewritten in the response. Default value: None
content_type	str	No	Explanation: Content-Type is rewritten in the response. Default value: None
expires	str	No	Explanation: Expires is rewritten in the response. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is obtained.

Table 8-74 GetObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
range	str	No	Explanation: Download range. For example, 0-999 indicates the download range is from byte 1 to byte 1,000. Value range: Value range: 0 to the object length minus 1. Format: <i>x-y</i>, indicating the range is from byte <i>x</i>+1 to byte <i>y</i>+1 Restrictions: The upper limit 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_modified_since	str or DateTime	No	Explanation: The object is returned if it has been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-75 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_unmodified_since	str or DateTime	No	Explanation: The object is returned if it has not been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-75 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: Each origin can contain only one wildcard character (*). Default value: None
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Table 8-75 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

List of returned results

Table 8-76 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-77 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-78 GetResult.body

GetResult.body Type	Description
ObjectStream	Explanation: Response to the request for downloading an object

Table 8-79 ObjectStream

Parameter	Type	Description
response	object	Explanation: If loadStreamInMemory is set to False and downloadPath is left blank in the request, this parameter is returned and indicates a readable stream. You can read object content from it. Default value: None
buffer	object	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the data stream in the memory. Default value: None
size	int	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the size of the data stream. Value range: An integer greater than or equal to 0, in bytes Default value: None
url	str	Explanation: If loadStreamInMemory is set to False and downloadPath is not left blank, this parameter is returned and indicates the download path. Default value: None
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. Default value: False

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-80. Default value: None
accessContorlAllo- wOrigin	str	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 the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com. Default value: None
accessContorlAllow- Headers	str	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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlAllow-Methods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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
accessControlExpose-Headers	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlMaxAge	int	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 only one MaxAgeSeconds . Value range: An integer greater than or equal to 0, in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
cacheControl	str	Explanation: Cache-Control header in the response Default value: None
contentDisposition	str	Explanation: Content-Disposition header in the response Default value: None
contentEncoding	str	Explanation: Content-Encoding header in the response Default value: None
contentType	str	Explanation: Content-Type header in the response Default value: None

Parameter	Type	Description
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
expires	str	Explanation: Expires header in the response Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT. Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: Default value: None

Parameter	Type	Description
expiration	str	Explanation: Expiration details. Example: " expiry-date= \\"Mon, 11 Sep 2023 00:00:00 GMT\\"" Default value: None
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL http://www.example.com/ : 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 for objects in the root directory of a bucket. Default value: None

Table 8-80 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example downloads a file.

```
from obs import GetObjectHeader
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional headers of the request for downloading an object.
    headers = GetObjectHeader()
    # (Optional) The object is returned if it has been modified since the specified time; otherwise, an error is
    # returned.
    headers.if_modified_since = 'date'
    # Specify the full path (localfile as an example) to which objects are downloaded. The full path contains
    # the local file name.
    downloadPath = 'localfile'
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Perform the file-based download.
    resp = obsClient.getObject(bucketName, objectKey, downloadPath, headers=headers)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Object Succeeded')
```

```
print('requestId:', resp.requestId)
print('url:', resp.body.url)
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

NOTE

- Use **downloadPath** to specify the path for saving the file to be downloaded. (If you choose the current directory, it must be contained in the path, for example, `./xxx`. The path cannot contain only the file name `xxx`.)
- If **loadStreamInMemory** is set to **True**, **downloadPath** is omitted.

NOTICE

If the value of **downloadPath** is a file path instead of a folder path, the object is downloaded as the file.

8.13 Downloading an Object - Range-Based (SDK for Python)

Function

This API downloads partial data of an object by specifying a range. If the specified range is from **0** to **1000**, data from byte 0 to byte 1,000, 1,001 bytes in total, are returned. If the specified range is invalid, data of the whole object is returned.

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).
- Objects in the storage class can be downloaded only when they are restored.
- If the specified range is invalid (for example, the start or end position is set to a negative integer or the range is larger than the object length), the whole object is returned.

Method

```
ObsClient.getObject(bucketName, objectKey, downloadPath, getObjectRequest, headers,
loadStreamInMemory, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-81 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
downloadPath	str	No	Explanation: The download path with the file name contained, for example, aa/bb.txt. Default value: None
getObjectRequest	GetObjectRequest	No	Explanation: Additional parameters of an object download request Value range: See Table 8-82.
headers	GetObjectHeader	No	Explanation: Additional headers of an object download request Value range: See Table 8-83.
loadStreamInMemory	bool	No	Explanation: Whether to load the data stream of the object to memory Value range: True: The obtained data stream is loaded to memory with downloadPath ignored. False: The obtained data stream is not loaded to memory. Default value: False

Parameter	Type	Mandatory (Yes/No)	Description
progressCallback	callable	No	Explanation: Callback function for obtaining the download progress Default value: None NOTE This function contains the following parameters in sequence: number of downloaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Downloading an Object - Obtaining the Download Progress (SDK for Python) .
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-82 GetObjectRequest

Parameter	Type	Mandatory (Yes/No)	Description
cache_control	str	No	Explanation: Cache-Control is rewritten in the response. Default value: None
content_disposition	str	No	Explanation: Content-Disposition is rewritten in the response. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
content_encoding	str	No	Explanation: Content-Encoding is rewritten in the response. Default value: None
content_language	str	No	Explanation: Content-Language is rewritten in the response. Default value: None
content_type	str	No	Explanation: Content-Type is rewritten in the response. Default value: None
expires	str	No	Explanation: Expires is rewritten in the response. Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is obtained.

Table 8-83 GetObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
range	str	No	Explanation: The object content within the specified range is downloaded. If the range value is invalid, the entire object is downloaded. Value range: The upper limit 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. Restrictions: - The value of range ranges from 0 to the object size minus 1. The start value of range is mandatory. If only the start value is specified, the object content from the start value to the object size minus 1 is downloaded. bytes=byte_range Example 1: bytes=0-4 Example 2: bytes=1024 Example 3: bytes=10-20,30-40 (multiple ranges) - Even when the range header is included, the ETag value in the response still corresponds to the entire object, not just the specified range. Default value: None
if_match	str	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. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_modified_since	str or Datetime	No	Explanation: The object is returned if it has been modified since the specified time; otherwise, an error is returned. Restrictions: You can generate a date time by referring to Table 8-84. For example, <code>Datetime(year=2023, month=9, day=12)</code> Default value: None
if_unmodified_since	str or Datetime	No	Explanation: The object is returned if it has not been modified since the specified time; otherwise, an error is returned. Restrictions: You can generate a date time by referring to Table 8-84. For example, <code>Datetime(year=2023, month=9, day=12)</code> Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: Each origin can contain only one wildcard character (*). Default value: None
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request. Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Table 8-84 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None

Parameter	Type	Description
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

List of returned results

Table 8-85 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-86 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None

Parameter	Type	Description
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-87 GetResult.body

GetResult.body Type	Description
ObjectStream	Explanation: Response to the request for downloading an object

Table 8-88 ObjectStream

Parameter	Type	Description
response	object	Explanation: If loadStreamInMemory is set to False and downloadPath is left blank in the request, this parameter is returned and indicates a readable stream. You can read object content from it.
buffer	object	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the data stream in the memory.

Parameter	Type	Description
size	int	Explanation: If loadStreamInMemory is set to True in the request, this parameter is returned and indicates the size of the data stream. Value range: 0 to $(2^{63} - 1)$, in bytes
url	str	Explanation: If loadStreamInMemory is set to False and downloadPath is not left blank, this parameter is returned and indicates the download path.
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. Default value: False
storageClass	str	Explanation: Object storage class. Value range: • If the storage class is Standard, leave this parameter blank. • For details about the available storage classes, see Table 8-89. Default value: None

Parameter	Type	Description
accessControlAllowedOrigin	str	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 the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com. Default value: None
accessControlAllowedHeaders	str	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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
accessControlAllowedMethods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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

Parameter	Type	Description
accessControlExposeHeaders	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
accessControlMaxAge	int	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 only one MaxAgeSeconds. Value range: 0 to ($2^{31} - 1$), in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
cacheControl	str	Explanation: Cache-Control header in the response Default value: None

Parameter	Type	Description
contentDisposition	str	Explanation: Content-Disposition header in the response Default value: None
contentEncoding	str	Explanation: Content-Encoding header in the response Default value: None
contentLanguage	str	Explanation: Content-Language header in the response Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
expires	str	Explanation: Expires header in the response Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT . Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: Default value: None

Parameter	Type	Description
expiration	str	Explanation: Expiration details. Example: " expiry-date= \\"Mon, 11 Sep 2023 00:00:00 GMT\\"" Default value: None
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation: anotherPage.html The request is redirected to an external URL http://www.example.com/ : 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 for objects in the root directory of a bucket. Default value: None

Table 8-89 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example downloads part of object **objectname** by specifying a range from 0 to 1,000. You can also specify multiple ranges when obtaining the **objectname** object.

```
from obs import GetObjectHeader
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional headers of the request for downloading an object.
    headers = GetObjectHeader()
    # Specify the range (0-1000 in this example) of the object bytes to be returned.
    # You can specify multiple ranges when obtaining an object. For example,
headers.range='0-1000,2000-3000'.
    headers.range = '0-1000'
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Perform the range download. If loadStreamInMemory is set to True, downloadpath will be invalid,
    and data streams will be downloaded to the memory.
    resp = obsClient.getObject(bucketName, objectKey, loadStreamInMemory=True, headers=headers)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
```

```
if resp.status < 300:
    print('Get Object Succeeded')
    print('requestId:', resp.requestId)
    # Obtain the object content.
    print('buffer:', resp.body.buffer)
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

NOTE

- If the specified range is invalid (for example, the start or end position is set to a negative integer or the range is larger than the object length), the whole object is returned.

8.14 Downloading an Object - Resumable (SDK for Python)

Function

Downloading large files often fails due to unstable network or program breakdown. It is a waste of resources to download files again. Moreover, the restarted download may still fail due to an unstable network. To resolve such issues, the resumable download API splits the file to be downloaded into multiple parts and downloads them separately. The download result of each part is recorded in a checkpoint file in real time. Only when all parts are downloaded is a message indicating the download is successful returned. If any parts fail to be downloaded, a message is returned telling you to call the API again to download the failed parts. Since the checkpoint file contains the progress of all parts, it helps you avoid downloading all parts in re-downloads, so that you can enjoy a cost-effective, efficient download.

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).
- The resumable download is an encapsulated and enhanced version of [partial download](#).
- This API saves resources and improves efficiency upon the re-download, and speeds up the download process by concurrently downloading 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 downloads of parts.
- **EnableCheckpoint**: The default value is **False**, indicating that resumable download is disabled. In this case, the resumable download API is a simple encapsulation of the partial download API, and no checkpoint file will be generated.
- **CheckpointFile**: This parameter is valid only when **EnableCheckpoint** is **True**.

Method

```
ObsClient.downloadFile(bucketName, objectKey, downloadFile, partSize, taskNum, enableCheckpoint,  
checkpointFile, header, versionId, progressCallback, extensionHeaders)
```

Request Parameters

Table 8-90 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
downloadFile	str	Yes	Explanation: Full local path for saving the file to be downloaded Default value: None
partSize	int	No	Explanation: Part size Value range: The value must be greater than 0 but less than the object size, in bytes. Default value: 5MB
taskNum	int	No	Explanation: Maximum number of parts that can be downloaded concurrently in a multipart download Value range: The value must be greater than 0 but not exceed the result of the file size divided by the part size (rounded up). Default value: 1, indicating concurrent downloads are not used.

Parameter	Type	Mandatory (Yes/No)	Description
enableCheckpoint	bool	No	Explanation: Whether to enable the resumable download mode Value range: True: The resumable download mode is enabled. False: The resumable download mode is disabled. Default value: False
checkpointFile	str	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 for resumable uploads. Default value: If this parameter is left blank, the checkpoint file will be saved in the current directory.
header	GetObjectHeader	No	Explanation: Headers in the request used for obtaining the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-91. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 Value range: The value must contain 32 characters. Default value: None. If this parameter is left blank, the latest version of the object is obtained.
progressCallback	callable	No	Explanation: Callback function for obtaining the download progress Default value: None NOTE This function contains the following parameters in sequence: number of downloaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Downloading an Object - Obtaining the Download Progress (SDK for Python).
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-91 GetObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
range	str	No	Explanation: Download range. For example, 0-999 indicates the download range is from byte 1 to byte 1,000. Value range: Value range: 0 to the object length minus 1. Format: x-y, indicating the range is from byte x+1 to byte y+1 Restrictions: The upper limit 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. Default value: None
if_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_modified_since	str or DateTime	No	Explanation: The object is returned if it has been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-92 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
if_unmodified_since	str or DateTime	No	Explanation: The object is returned if it has not been modified since the specified time; otherwise, an error is returned. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-92 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: Each origin can contain only one wildcard character (*). Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request Only CORS requests matching the allowed headers are valid. Restrictions: Each header can contain only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Table 8-92 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0

Parameter	Type	Description
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

Responses

Table 8-93 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-94 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-95 GetResult.body

GetResult.body Type	Description
GetObjectMetadataResponse	Explanation: For details, see Table 8-96 .

Table 8-96 GetObjectMetadataResponse

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-97. Default value: None
accessControlAllowOrigin	str	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 the requests can access the bucket. Restrictions: Domain name of the origin. Each origin can contain only one wildcard character (*), for example, https://*.vbs.example.com . Default value: None

Parameter	Type	Description
accessControlAllow-Headers	str	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 only one wildcard character (*). Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
accessControlAllow-Methods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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

Parameter	Type	Description
accessControlExposeHeaders	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
accessControlMaxAge	int	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 only one MaxAgeSeconds. Value range: An integer greater than or equal to 0, in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None

Parameter	Type	Description
contentType	str	Explanation: MIME type of the object to be downloaded. 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)? (Python SDK) Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT. Default value: None
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: This parameter is only available for objects. Default value: None
expiration	str	Explanation: Expiration details. Example: "expiry-date= \"Mon, 11 Sep 2023 00:00:00 GMT\"" Default value: None

Parameter	Type	Description
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation:/anotherPage.html The request is redirected to an external URL http://www.example.com/: 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 for objects in the root directory of a bucket. Default value: None
isAppendable	bool	Explanation: Whether the object is appendable Value range: True: The object is appendable. False: The object is not appendable. Default value: None
nextPosition	int	Explanation: Start position for next appending Value range: 0 to the object length, in bytes. Default value: None

Table 8-97 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example downloads object **objectname** from bucket **examplebucket** using resumable download.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the full path to which objects are downloaded. The full path contains the local file name.
    downloadFile = 'localfile'
```

```
# Specify the number of parts that can be concurrently downloaded.
taskNum = 5
# Specify the part size.
partSize = 10 * 1024 * 1024
# Enable the resumable download by setting enableCheckpoint to True.
enableCheckpoint = True
# Download the object using resumable download.
resp = obsClient.downloadFile(bucketName, objectKey, downloadFile, partSize, taskNum,
enableCheckpoint)

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Download File Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Download File Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Download File Failed')
    print(traceback.format_exc())
```

8.15 Downloading an Object - Obtaining the Download Progress (SDK for Python)

You can obtain the download progress when downloading an object in binary, streaming, file-based, or resumable mode.

This example returns the object download progress.

Sample code is as follows:

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

# Obtain the download progress.
def callback(transferredAmount, totalAmount, totalSeconds):
    # Obtain the average download rate (KB/s).
    print(transferredAmount * 1.0 / totalSeconds / 1024)
    # Obtain the download progress in percentage.
    print(transferredAmount * 100.0 / totalAmount)

try:
    bucketName="examplebucket"
    objectKey="objectname"
    # Download an object.
    resp = obsClient.getObject(bucketName=bucketName,objectKey=objectKey, progressCallback=callback)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
```



```
if resp.status < 300:
    print('Get Object Succeeded')
    print('requestId:', resp.requestId)
    # Read the object content.
    while True:
        chunk = resp.body.response.read(65536)
        if not chunk:
            break
        print(chunk)
    resp.body.response.close()
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

8.16 Uploading an Object - Creating a Folder (SDK for Python)

Function

This API creates a folder in an existing bucket to manage data in OBS.

OBS does not involve folders like in a file system. All elements stored in OBS buckets are objects. To create a folder in OBS is essentially to create an object whose size is 0 and whose name ends with a slash (/). Such objects are no different from other objects (you can perform normal operations on them such as download and delete), except that they are displayed as folders in OBS console.

Restrictions

- To create a folder, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy).
- To create a folder in OBS is essentially to create an object whose size is 0 and whose name ends with a slash (/).
- If the folder you create is a subfolder, its parent folders will not be automatically created. For example, if you call this API to create folder **src1/src2/src3/**, this folder will be created, but **src1/** and **src1/src2/** will not.

Method

```
ObsClient.putContent(bucketName, objectKey, content, metadata, headers, progressCallback, autoClose,
extensionHeaders)
```

Request Parameters

Table 8-98 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
content	str or readable object	No	Explanation: Content of the object to be uploaded Value range: None Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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
headers	PutObject Header	No	Explanation: Headers in the request used for configuring the storage class, redundancy policy, and other basic information about the object Value range: See Table 8-99. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This function contains the following parameters in sequence: number of uploaded bytes, total number of bytes, and used time (in seconds). For details about the sample code, see Obtaining the Upload Progress (SDK for Python). Streaming uploads, file-based uploads, multipart uploads, appendable uploads, and resumable uploads are supported.
autoClose	bool	No	Explanation: Whether or not to automatically close data streams after the upload is complete Value range: True: The data stream is automatically closed. False: The data stream is not automatically closed. Default value: True
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 8-99 PutObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the data to be uploaded. It is used for the OBS server to verify data integrity. Value range: Base64-encoded 128-bit MD5 value of the request body calculated according to RFC 1864 Example: n58IG6hfM7vqI4K0vnWpog== Default value: None
acl	str	No	Explanation: ACL that can be pre-defined during object creation Value range: For details about the available options, see HeadPermission . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: If you do not specify contentType 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 contentType.

Parameter	Type	Mandatory (Yes/No)	Description
contentLength	int	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, multipart uploads should be used. Default value: If this parameter is not specified, OBS SDK for Python automatically calculates the size of the object.
storageClass	str	No	Explanation: Storage class of the object Value range: See Table 8-101. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. - If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object to be uploaded Value range: See Table 8-102. Default value: None
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None

Table 8-100 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-101 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-102 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeld	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-103 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Responses

Table 8-104 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-105 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-106 GetResult.body

GetResult.body Type	Description
PutContentResponse	Explanation: Response to the request for uploading an object. For details, see Table 8-10 .

Table 8-107 PutContentResponse

Parameter	Type	Description
storageClass	str	Explanation: Storage class of the object Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-101. Default value: None
versionId	str	Explanation: Object version ID. If versioning is enabled for the bucket, the object version ID will be returned. Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
objectUrl	str	Explanation: Full path to the object Default value: None

Code Examples

This example creates a folder named **parent_directory/**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
```



```
bucketName = "examplebucket"
# Specify a folder name ending with a slash (/).
objectKey = "parent_directory/"
# Create a folder object whose name ends with a slash (/). To avoid unexpected charges, do not upload
files to the folder during creation.
resp = obsClient.putContent(bucketName, objectKey, content=None)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Put Content Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Put Content Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

8.17 Copying an Object (SDK for Python)

Function

This API copies an object stored in OBS to another path, with a copy created during the process. You can create a copy of an object 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).
- The object copy request carries the information about the source bucket and object to be copied in the header field. The message body cannot be carried.
- The target object size ranges from 0 to 5 GB. If the source object size exceeds 5 GB, you can only copy parts of the object by referring to [APIs Related to Multipart Upload \(SDK for Python\)](#).

Method

```
ObsClient.copyObject(sourceBucketName, sourceObjectKey, destBucketName, destObjectKey, metadata,
headers, versionId, extensionHeaders)
```

Request Parameters

Table 8-108 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
sourceBucketName	str	Yes	Explanation: Name of the source bucket 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
sourceObjectKey	str	Yes	Explanation: Source 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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	No	Explanation: Version ID of the source object, for example, G001117FCE89978B0000401205D5DC9A Value range: The value must contain 32 characters. Default value: If this parameter is left blank, the latest version of the source object is copied.

Parameter	Type	Mandatory (Yes/No)	Description
destBucketName	str	Yes	Explanation: Target 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
destObjectKey	str	Yes	Explanation: Target 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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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: - You need to set directive in headers to REPLACE. - The custom metadata cannot exceed 8 KB. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. - 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
headers	CopyObjectHeader	No	Explanation: Request headers for copying an object. For details, see Table 8-109. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-109 CopyObjectHeader

Parameter	Type	Mandatory (Yes/No)	Description
acl	str	No	Explanation: ACL that can be pre-defined during object copying Default value: None
directive	str	No	Explanation: Whether to copy source object attributes to the target object Value range: COPY (attributes of the target object are copied from the source object) REPLACE (attributes of the target object are replaced with values specified in the request parameters) Default value: COPY

Parameter	Type	Mandatory (Yes/No)	Description
if_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_none_match	str	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. Value range: The value must contain 32 characters. Default value: None
if_modified_since	str or DateTime	No	Explanation: The source object is copied if it has been modified since the specified time; otherwise, an exception is thrown. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-112 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
if_unmodified_since	str or DateTime	No	Explanation: The source object is copied if it has not been modified since the specified time; otherwise, an exception is thrown. Restrictions: The value must be in the GMT format. For example, Wed, 25 Mar 2020 02:39:52 GMT. You can refer to Table 8-112 to specify time. For example, DateTime(year=2023, month=9, day=12) Default value: None
location	str	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: location:/anotherPage.html The request is redirected to an external URL http://www.example.com/: location:http://www.example.com/ OBS obtains the specified value from the header and stores it in the object metadata location. Restrictions: • The value must start with a slash (/), http://, or https:// and cannot exceed 2 KB. • OBS only supports redirection for objects in the root directory of a bucket. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
cacheControl	str	No	Explanation: Cache-Control is rewritten in the response. Default value: None
contentDisposition	str	No	Explanation: Content-Disposition is rewritten in the response. Default value: None
contentEncoding	str	No	Explanation: Content-Encoding is rewritten in the response. Default value: None
contentLanguage	str	No	Explanation: Content-Language is rewritten in the response. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
expires	str	No	Explanation: Expiration details. Example: " expiry-date= \\"Mon, 11 Sep 2023 00:00:00 GMT\\"" Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	No	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-111. Default value: None
successActionRedirect	str	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 in the returned results contains SuccessActionRedirect as well as the bucket name, object name, and object ETag. - If the value is invalid, OBS ignores this parameter. In such case, Location in the returned results indicates the object address, and OBS returns a status code based on whether the operation succeeds or fails. Default value: None
extensionGrants	list of ExtensionGrant	No	Explanation: List of the extended permissions for the object Value range: See Table 8-113. Default value: None

Table 8-110 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-111 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.

Parameter	Type	Description
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 8-112 DateTime

Parameter	Type	Description
year	int	Explanation: Year in UTC Default value: None
month	int	Explanation: Month in UTC Default value: None
day	int	Explanation: Day in UTC Default value: None
hour	int	Explanation: Hour in UTC Restrictions: The value is in 24-hour format. Default value: 0
min	int	Explanation: Minute in UTC Default value: 0
sec	int	Explanation: Second in UTC Default value: 0

Table 8-113 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeld	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions Default value: None

Table 8-114 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.

Constant	Description
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Responses

Table 8-115 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-116 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-117 GetResult.body

GetResult.body Type	Description
CopyObjectResponse	Explanation: Response to the request for copying an object

Table 8-118 CopyObjectResponse

Parameter	Type	Description
lastModified	str	Explanation: Time when the last modification was made to the target object Value range: UTC time Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
copySourceVersionId	str	Explanation: Version ID of the source object. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Version ID of the target object Value range: The value must contain 32 characters. Default value: None

Code Examples

This example copies an object from one bucket to another.

```
from obs import ObsClient
from obs import CopyObjectHeader
import os
import traceback
```

```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify additional headers of the request for copying an object.
    headers = CopyObjectHeader()
    # Set directive of headers to REPLACE when custom metadata is configured.
    headers.directive = 'REPLACE'
    # Specify the value of if_match. If the ETag of the source object is the same as the one specified by this
parameter, the source object is copied. Otherwise, an error code is returned.
    headers.if_match = '8f4e*****ba8c'
    # Specify custom metadata of the target object.
    metadata = {'meta1': 'value1'}
    # Specify a source bucket.
    sourceBucketName = 'sourcebucket'
    # Specify a source object.
    sourceObjectKey = 'sourceobjectkey'
    # Specify a target bucket.
    destBucketName = 'destbucket'
    # Specify a target object name.
    destObjectKey = 'destobjectkey'
    # Copy the object.
    resp = obsClient.copyObject(sourceBucketName, sourceObjectKey, destBucketName, destObjectKey,
                                metadata, headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Copy Object Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
        print('lastModified:', resp.body.lastModified)
        print('versionId:', resp.body.versionId)
        print('copySourceVersionId:', resp.body.copySourceVersionId)
    else:
        print('Copy Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Copy Object Failed')
    print(traceback.format_exc())
```

8.18 Deleting an Object (SDK for Python)

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).

- If versioning is not enabled for a bucket, deleted objects cannot be recovered.

Method

```
ObsClient.deleteObject(bucketName, objectKey, versionId, extensionHeaders)
```

Request Parameters

Table 8-119 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	Yes	Explanation: Target 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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 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.
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 8-120 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-121 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-122 GetResult.body

GetResult.body Type	Description
DeleteObjectResponse	Explanation: Response results of the request for deleting an object For details, see Table 8-123 .

Table 8-123 DeleteObjectResponse

Parameter	Type	Description
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. Default value: false
versionId	str	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 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.

Code Examples

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

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify versionId to delete an object version. If versioning is not enabled, do not specify that
    # parameter.
    # versionId = 'G001119522F0D8Cxxxxxxxxxxxxxxxxxxxx'
    # Delete the object.
    resp = obsClient.deleteObject(bucketName, objectKey)
    # If the return code is 2xx, the API is successfully called. Otherwise, the API fails to be called. If a
```



```
nonexistent version ID is used to delete a versioned object (that is, the requested object does not exist), a success message is still returned.
if resp.status < 300:
    print('Delete Object Succeeded')
    print('requestId:', resp.requestId)
    print('deleteMarker:', resp.body.deleteMarker)
    print('versionId:', resp.body.versionId)
else:
    print('Delete Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Delete Object Failed')
    print(traceback.format_exc())
```

8.19 Batch Deleting Objects (SDK for Python)

Function

This API deletes objects in a batch from a specific bucket. Deleted objects cannot be restored.

OBS does not involve folders like in a file system. All elements stored in OBS buckets are objects. A folder you see on the console or other tools in OBS is essentially an object whose size is 0 and whose name ends with a slash (/). To delete a folder, you need to list all objects whose names are prefixed with the folder name and then call the batch deletion API.

In a batch delete operation, 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).
- If versioning is not enabled for a bucket, deleted objects cannot be recovered.
- 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(bucketName, deleteObjectsRequest, extensionHeaders)
```

Request Parameters

Table 8-124 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set

Parameter	Type	Mandatory (Yes/No)	Description
			in the first creation request. Default value: None
deleteObjectsRequest	DeleteObjectsRequest	Yes	Explanation: Request parameters of deleting objects in a batch For details, see Table 8-125 . Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-125 DeleteObjectsRequest

Parameter	Type	Mandatory (Yes/No)	Description
quiet	bool	No	Explanation: Response mode to the request for deleting objects in a batch 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

Parameter	Type	Mandatory (Yes/No)	Description
objects	list of Object	Yes	Explanation: List of objects to be deleted. For details, see Table 8-126 . Default value: None

Table 8-126 Object

Parameter	Type	Mandatory (Yes/No)	Description
key	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 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 8-127 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-128 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None

Parameter	Type	Description
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-129 GetResult.body

GetResult.body Type	Description
DeleteObjectResponse	Explanation: Response results of the request for deleting objects in a batch For details, see Table 8-130 .

Table 8-130 DeleteObjectResponse

Parameter	Type	Description
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. Default value: false
versionId	str	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9 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.

Code Examples

This example deletes objects **objectkey1** and **objectkey2** from bucket **examplebucket** in a batch.

```
from obs import ObsClient
import os
from obs import DeleteObjectsRequest
from obs import Object
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the objects to be deleted in a batch.
    object1 = Object(key='objectkey1', versionId=None)
    object2 = Object(key='objectkey2', versionId=None)

    # Specify encoding_type when the object name contains special characters.
```

```
encoding_type = 'url'
bucketName = "examplebucket"
# Batch delete the objects.
resp = obsClient.deleteObjects(bucketName, DeleteObjectsRequest(quiet=False, objects=[object1,
object2],
                                encoding_type=encoding_type))

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Delete Objects Succeeded')
    print('requestId:', resp.requestId)
    if resp.body.deleted:
        index = 1
        for delete in resp.body.deleted:
            print('delete[' + str(index) + ']')
            print('key:', delete.key, 'deleteMarker:', delete.deleteMarker, 'deleteMarkerVersionId:',
                delete.deleteMarkerVersionId)
            print('versionId:', delete.versionId)
            index += 1
    if resp.body.error:
        index = 1
        for err in resp.body.error:
            print('err[' + str(index) + ']')
            print('key:', err.key, 'code:', err.code, 'message:', err.message)
            print('versionId:', err.versionId)
            index += 1
    else:
        print('Delete Objects Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Objects Failed')
    print(traceback.format_exc())
```

This example deletes all objects prefixed with **test/** in bucket **examplebucket**.



WARNING

In the example below, if the value of **prefix** is an empty string or NULL, all files in the bucket will be deleted.

```
from obs import ObsClient
import os
from obs import DeleteObjectsRequest
from obs import Object
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Specify the folder to be deleted.
```



```
prefix = 'test/'
# Specify the maximum number of objects to be listed at a time. 1000 is used in this example.
max_num = 1000
mark = None
index = 1
failed_list = []
while True:
    resp = obsClient.listObjects(bucketName=bucketName, prefix=prefix, marker=mark,
max_keys=max_num,
                                encoding_type='url')
    if resp.status < 300:
        need_to_delete_objects = [Object(key=i["key"], versionId=None) for i in resp.body["contents"]]
        del_resp = obsClient.deleteObjects(bucketName,
                                DeleteObjectsRequest(False, need_to_delete_objects, encoding_type="url"))
        for delete in del_resp.body.deleted:
            print("Successfully deleted %s " % delete.key)
            index += 1
        if del_resp.body.error:
            for err in del_resp.body.error:
                print("Failed to delete %s" % err.key)
                failed_list.append(err.key)
        if resp.body.is_truncated is True:
            mark = resp.body.next_marker
        else:
            break
    else:
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
        break
print("Total deleted %s objects" % index)
for i in failed_list:
    print("Failed to delete %s, please try again" % i)
except:
    print('Delete Objects Failed')
    print(traceback.format_exc())
```

8.20 Obtaining Object Metadata (SDK for Python)

Function

Object metadata is a set of name-value pairs that describe the object and is used for object management.

Currently, only the system-defined metadata can be obtained. System-defined metadata consists of system-controlled metadata and user-controlled metadata.

This API sends a HEAD request to obtain the metadata of an object in a specified bucket.

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).
- Users with the READ permission on objects can send HEAD requests to retrieve metadata from objects. The metadata of the objects is included in the response.

Method

```
ObsClient.getObjectMetadata(bucketName, objectKey, versionId, origin, requestHeaders, extensionHeaders)
```

Request Parameters

Table 8-131 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Value range: The value must contain 32 characters. Default value: None
origin	str	No	Explanation: Origin of the cross-domain request specified by the preflight request. Generally, it is a domain name. Restrictions: You can configure one or more rules and use at most one wildcard character (*) in each rule. If you want to configure multiple rules, separate them using a line breaker. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
requestHeaders	str	No	Explanation: HTTP headers in a cross-origin request. Only CORS requests matching the allowed headers are valid. Restrictions: You can enter multiple allowed headers, with one separated from another using a line break. Each header can contain one wildcard character (*) at most. Spaces, ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 8-132 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-133 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-134 GetResult.body

GetResult.body Type	Description
GetObjectMetadataResponse	Explanation: Response to the request for obtaining object metadata For details, see Table 8-135 .

Table 8-135 GetObjectMetadataResponse

Parameter	Type	Description
storageClass	str	Explanation: Object storage class. Value range: - If the storage class is Standard, leave this parameter blank. - For details about the available storage classes, see Table 8-136. Default value: None
accessControlAllowOrigin	str	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 only one wildcard character (*), for example, https://*.vbs.example.com . Default value: None
accessControlAllowHeaders	str	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
accessControlAllow-Methods	str	Explanation: AllowedMethod in the CORS rules of the bucket. It specifies the HTTP method of 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
accessControlExpose-Headers	str	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, wildcard characters (*), ampersands (&), colons (:), and less-than signs (<) are not allowed. Default value: None

Parameter	Type	Description
accessControlMaxAge	int	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 only one MaxAgeSeconds. Value range: An integer greater than or equal to 0, in seconds Default value: 100
contentLength	int	Explanation: Object size Value range: The value ranges from 0 TB to 48.8 TB, in bytes. Default value: None
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT. Default value: None

Parameter	Type	Description
etag	str	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 is B when the object is downloaded, this indicates the contents of the object are changed. The ETag reflects changes only to the contents of an object, 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. Value range: The value must contain 32 characters. Default value: None
versionId	str	Explanation: Object version ID. Value range: The value must contain 32 characters. Default value: None
restore	str	Explanation: Restore status of an object. This header is returned when object is being restored or has been restored. 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. expiry-date indicates when the restored object expires. Restrictions: This parameter is only available for objects. Default value: None

Parameter	Type	Description
expiration	str	Explanation: Expiration details. Example: "expiry-date=\`Mon, 11 Sep 2023 00:00:00 GMT\`" Default value: None
websiteRedirectLocation	str	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. This parameter specifies the address the request for the object is redirected to. The request is redirected to object anotherPage.html in the same bucket: WebsiteRedirectLocation: anotherPage.html The request is redirected to an external URL http://www.example.com/: 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 for objects in the root directory of a bucket. Default value: None
isAppendable	bool	Explanation: Whether the object is appendable Value range: True: The object is appendable. False: The object is not appendable. Default value: None

Parameter	Type	Description
nextPosition	int	Explanation: Start position for next appending Value range: 0 to the object length, in bytes. Default value: None

Table 8-136 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

 NOTE

You can obtain custom metadata from the response headers.

Code Examples

This example returns the metadata of object **objectname**.

```
from obs import ObsClient
import os
import traceback
```

```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Obtain object metadata.
    resp = obsClient.getObjectMetadata(bucketName, objectKey)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Object Metadata Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
        print('lastModified:', resp.body.lastModified)
        print('contentType:', resp.body.contentType)
        print('contentLength:', resp.body.contentLength)
    else:
        print('Get Object Metadata Failed')
        print('requestId:', resp.requestId)
        print('status:', resp.status)
        print('reason:', resp.reason)
except:
    print('Get Object Metadata Failed')
    print(traceback.format_exc())
```

8.21 Modifying Object Metadata (SDK for Python)

Function

Object metadata is a set of name-value pairs that describe the object and is used for object management.

Currently, only the system-defined metadata can be modified.

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 modifies, deletes, or adds metadata to 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).
- If versioning is enabled for a bucket, you can set metadata for objects of the latest version, but cannot set metadata for objects of historical versions.

- You cannot set metadata for objects.

Method

ObsClient.setObjectMetadata(bucketName, objectKey, metadata, headers, versionId, extensionHeaders)

Request Parameters

Table 8-137 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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

Parameter	Type	Mandatory (Yes/No)	Description
headers	SetObjectMetadataHeader	No	Explanation: Request headers for configuring object metadata. For details, see Table 8-138 .
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Value range: The value must contain 32 characters. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-138 SetObjectMetadataHeader

Parameter	Type	Mandatory (Yes/No)	Description
removeUnset	bool	No	Explanation: Used to set the metadata operation indicator Value range: True: The metadata operation indicator is REPLACE. REPLACE uses the complete header carried in the current request to replace the original one and deletes the metadata that is not specified. False: The metadata operation indicator is 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. Default value: False

Parameter	Type	Mandatory (Yes/No)	Description
location	str	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 http://www.example.com/: 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 for objects in the root directory of a bucket. Default value: None
cacheControl	str	No	Explanation: Cache-Control header in the response. It specifies the cache behavior of the web page when an object is downloaded. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
contentDisposition	str	No	Explanation: Content-Disposition header in the response It specifies the name of an object when it is downloaded. Default value: None
contentEncoding	str	No	Explanation: Content-Encoding header in the response It specifies the content encoding format when an object is downloaded. Default value: None
contentLanguage	str	No	Explanation: Content-Language header in the response It specifies the content language format when an object is downloaded. Default value: None
contentType	str	No	Explanation: Content-Type header in the response. It specifies the file type of an object when it is downloaded. Default value: None
expires	str	No	Explanation: Expires header in the response. It specifies the cache expiration time of the web page when the object is downloaded. CAUTION This parameter cannot be used to configure the expiration time of an object. To configure the object expiration time, see Setting an Object Expiration Time (SDK for Python) . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
storageClass	str	No	Explanation: Storage classes Value range: See Table 8-139 . Default value: None

Table 8-139 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Responses

Table 8-140 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-141 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example configures the object metadata.

```
from obs import SetObjectMetadataHeader
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
```

```
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify custom metadata.
    metadata = {'property1': 'property-value1', 'property2': 'property-value2'}
    # Specify the additional headers of the request for configuring object metadata.
    headers = SetObjectMetadataHeader()
    # (Optional) Specify the MIME type of the object.
    headers.contentType = "Your Content-Type"
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Configure metadata for the object.
    resp = obsClient.setObjectMetadata(bucketName, objectKey, metadata, headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Object Metadata Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Object Metadata Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Object Metadata Failed')
    print(traceback.format_exc())
```

8.22 Configuring an Object ACL (SDK for Python)

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. If an object is encrypted with SSE-KMS, the ACL configured for it is not in effect in the cross-tenant case.

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).
- An object can have a maximum of 100 rules in its ACL.

Method

```
ObsClient.setObjectAcl(bucketName, objectKey, acl, versionId, aclControl, extensionHeaders)
```

Request Parameters

Table 8-142 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
acl	ACL	No	Explanation: ACL of an object. Value range: See Table 8-143 . Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Value range: The value must contain 32 characters. Default value: None
aclControl	HeadPermission	No	Explanation: Pre-defined ACL. For details, see Table 8-144 . Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

 NOTE

`acl` and `aclControl` are mutually exclusive.

Table 8-143 ACL

Parameter	Type	Mandatory (Yes/No)	Description
owner	Owner	Yes if used as a request parameter	Explanation: Bucket owner For details, see Table 8-145 . Restrictions: Owner and Grants must be used together and they cannot be used with aclControl . Default value: None
grants	list of Grant	Yes if used as a request parameter	Explanation: List of grantees' permission information. For details, see Table 8-146 . Restrictions: Owner and Grants must be used together and they cannot be used with aclControl . Default value: None
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to objects in the bucket. This parameter is valid only when you configure the object ACL. Value range: True: The bucket ACL is applied to objects in the bucket. False: The bucket ACL is not applied to objects in the bucket. Default value: False

Table 8-144 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 8-145 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 8-146 Grant

Parameter	Type	Mandatory (Yes/No)	Description
grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee Value range: See Table 8-148 . Default value: None
permission	str	Yes if used as a request parameter	Explanation: Granted permission Value range: See Table 8-147 . Default value: None
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 8-147 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Table 8-148 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
grantee_id	str	Yes if the parameter is used as a request parameter and group is left blank	Explanation: Account (domain) ID of the grantee Value range: Default value: None
grantee_name	str	No if used as a request parameter	Explanation: Account name of the grantee Restrictions: • Cannot contain full-width characters. • Starts with a letter. • Contains 6 to 32 characters. • Contains only letters, digits, hyphens (-), and underscores (_). Default value: None
group	str	Yes if the parameter is used as a request parameter and grantee_id is left blank	Explanation: Authorized user group Value range: See Table 8-149 . Default value: None

Table 8-149 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Responses

Table 8-150 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-151 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Setting an Object ACL by Specifying acl

This example sets the ACL for object **objectname** to read and write for an IAM user (**userid**).

```
from obs import ObsClient
import os
from obs import ACL
from obs import Owner
from obs import Grant, Permission
from obs import Grantee
import traceback
```

```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the account ID of the bucket owner (ownerid as an example).
    owner = Owner(owner_id='ownerid')
    # Grant the read and write permissions to an IAM user (userid).
    grantee = Grantee(grantee_id='userid')
    grant0 = Grant(grantee=grantee, permission=Permission.READ)
    grant0 = Grant(grantee=grantee, permission=Permission.WRITE)
    # Set the ACL.
    acl = ACL(owner=owner, grants=[grant0])
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Configure the object ACL by specifying acl.
    resp = obsClient.setObjectAcl(bucketName, objectKey, acl)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Object Acl Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Object Acl Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Object Acl Failed')
    print(traceback.format_exc())
```

Setting an Object ACL by Specifying aclControl

This example sets a pre-defined object ACL to private read and write.

```
from obs import ObsClient
import os
from obs import HeadPermission
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Set the pre-defined ACL to PRIVATE to ensure high security.
```

```
aclControl = HeadPermission.PRIVATE
bucketName = "examplebucket"
objectKey = "objectname"
# Configure the object ACL by specifying acl.
resp = obsClient.setObjectAcl(bucketName, objectKey, aclControl=aclControl)

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Set Object Acl Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Set Object Acl Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Set Object Acl Failed')
    print(traceback.format_exc())
```

8.23 Obtaining an Object ACL (SDK for Python)

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.

This API obtains the ACL of an object in a specified bucket.

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).
- To call this API, you must have the read permission on the ACL of the object.

Method

```
ObsClient.getObjectAcl(bucketName, objectKey, versionId, extensionHeaders)
```

Request Parameters

Table 8-152 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
versionId	str	No	Explanation: Object version ID, for example, G001117FCE89978B0000401205D5DC9A Value range: The value must contain 32 characters. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 8-153 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 8-154 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 8-155 GetResult.body

GetResult.body Type	Description
ACL	Explanation: Object ACL. For details, see ACL . Default value: None

Table 8-156 ACL

Parameter	Type	Mandatory (Yes/No)	Description
owner	Owner	Yes if used as a request parameter	Explanation: Bucket owner. For details, see Table 8-157. Restrictions: owner and grants must be used together and they cannot be used with ACL. Default value: None
grants	list of Grant	Yes if used as a request parameter	Explanation: List of grantees' permission information. For details, see Table 8-158. Default value: None
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 8-157 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 8-158 Grant

Parameter	Type	Mandatory (Yes/No)	Description
grantee	Grantee	Yes if used as a request parameter	Explanation: Grantee information. For details, see Table 8-160 . Default value: None
permission	str	Yes if used as a request parameter	Explanation: Granted permission Value range: See Table 8-159 . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
delivered	bool	No if used as a request parameter	Explanation: Whether the bucket ACL is applied to all objects in the bucket Value range: True: The bucket ACL is applied to all objects in the bucket. False: The bucket ACL is not applied to all objects in the bucket. Default value: False

Table 8-159 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.

Constant	Description
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ, WRITE, READ_ACP, and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ, READ_ACP, and WRITE_ACP permissions for the object.

Table 8-160 Grantee

Parameter	Type	Mandatory (Yes/No)	Description
grantee_id	str	Yes if the parameter is used as a request parameter and group is left blank	Explanation: Account (domain) ID of the grantee Value range: Default value: None
grantee_name	str	No if used as a request parameter	Explanation: Account name of the grantee Restrictions: • Cannot contain full-width characters. • Starts with a letter. • Contains 6 to 32 characters. • Contains only letters, digits, hyphens (-), and underscores (_). Default value: None
group	str	Yes if the parameter is used as a request parameter and grantee_id is left blank	Explanation: Authorized user group Value range: See Table 8-161. Default value: None

Table 8-161 Group

Constant	Description
ALL_USERS	All users
AUTHENTICATED_USERS	Authorized users. This constant is deprecated.
LOG_DELIVERY	Log delivery group. This constant is deprecated.

Code Examples

This example returns the object ACL information of object **objectname**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Obtain the ACL of an object in a specified bucket.
    resp = obsClient.getObjectAcl(bucketName, objectKey)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Object Acl Succeeded')
        print('requestId:', resp.requestId)
        print('owner_id:', resp.body.owner.owner_id)
        print('owner_name:', resp.body.owner.owner_name)

        index = 1
        for grant in resp.body.grants:
            print('grant [' + str(index) + ']')
            print('grantee_id:', grant.grantee.grantee_id)
            print('grantee_name:', grant.grantee.grantee_name)
            print('group:', grant.grantee.group)
            print('permission:', grant.permission)
            index += 1
    else:
        print('Get Object Acl Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Acl Failed')
    print(traceback.format_exc())
```

8.24 Restoring a Cold Object (SDK for Python)

Function

To obtain the contents of an object in the Cold storage class, you need to restore the object first and then you can download it. After an object is restored, a copy of the object is saved in the Standard storage class. By doing so, the object in the Cold 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 expires.

This API is used to restore Cold objects in a specified bucket.

Restrictions

- To restore a Cold object, you must be the bucket owner or have the required permission (**obs:object:RestoreObject** in IAM or **RestoreObject** in a bucket policy.)
- To prolong the validity period of the Cold data restored, you can repeatedly restore the data, but you will be billed for each restore. 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.

Method

```
ObsClient.restoreObject(bucketName, objectKey, days, tier, versionId, extensionHeaders)
```

Request Parameters

Table 8-162 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of the same name in the same region, no error will be reported and the bucket properties comply with those set in the first creation request. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
days	int	Yes	Explanation: After an object is restored, a Standard copy of it is generated. This parameter specifies how long the Standard copy can be retained, that is, the validity period of the restored object. Value range: The value ranges from 1 to 30, in days. Default value: None
tier	str	No	Explanation: Retrieval speed tiers. You can select a suitable tier based on your requirements for retrieval speed. Value range: See Table 8-163. Default value: Standard
versionId	str	No	Explanation: Default value: None. If this parameter is left blank, the latest version of the object is specified.

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 8-163 RestoreTier

Constant	Description
Expedited	Objects can be restored at an expedited speed within 1 to 5 minutes.
Standard	Objects can be restored at a standard speed within 3 to 5 hours.

Responses

Table 8-164 List of returned results

Type	Description
GetResult	Explanation: SDK common results

NOTE

If **GetResult.status** is **202**, the object is being restored. If **GetResult.status** is **200**, the object has been restored.

Table 8-165 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example restores the Cold object **objectname**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify how long the restored object will be retained, in days. The value ranges from 1 to 30.
    days = 1
```

```
# Specify the restoration speed. Options: Expedited or Standard
tier = "Expedited"
# Restore the Cold object.
resp = obsClient.restoreObject(bucketName, objectKey, days, tier)

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Restore Object Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Restore Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Restore Object Failed')
    print(traceback.format_exc())
```

8.25 Using CRC64 to Verify File Consistency (SDK for Python)

NOTICE

- CRC calculation is a compute-intensive process that requires a large number of shift, exclusive OR (XOR), and table lookup operations. Enabling CRC consumes additional time, and the larger the data volume, the more CPU time is required. In C extension mode, calculating CRC for 1 GB of data takes about 4 to 6 seconds.
- The OBS SDK for Python requires `crcmod` to calculate CRC checksums, and `crcmod` depends on the **Python.h** file in the **python-devel** package. `crcmod` supports both the C extension mode and the Python mode. If there is no **Python.h** file in the system, the OBS SDK for Python can still be installed, but the C extension mode of `crcmod` will fail to be installed. In this case, when calculating CRC checksums during upload or download, the system will use the pure Python mode, whose performance and efficiency are far lower than those of the C extension mode. Therefore, before installing the OBS SDK for Python, ensure that **python-devel** has been installed. If it is not installed, disable CRC64 verification.

You can use the following APIs to verify file consistency using CRC64 during uploads:

- [Uploading Objects - Text-Based \(SDK for Python\)](#)
- [Uploading an Object - File-Based \(SDK for Python\)](#)
- [Uploading an Object - Append \(SDK for Python\)](#)
- [Uploading an Object - Resumable \(SDK for Python\)](#)
- [Multipart Upload Overview \(SDK for Python\)](#)

You can use the following APIs to verify file consistency using CRC64 during downloads:

- [Downloading an Object - File-Based \(SDK for Python\)](#)
- [Downloading an Object - Resumable \(SDK for Python\)](#)

- [Downloading an Object - Binary \(SDK for Python\)](#)
- [Downloading an Object - Streaming \(SDK for Python\)](#)

You can use the following APIs to obtain the CRC64 value of an object:

- [Downloading an Object - Range-Based \(SDK for Python\)](#)
- [Obtaining Object Metadata \(SDK for Python\)](#)

Sample Code 1: Performing a Text-Based Upload

This example uses CRC64 to verify file consistency during text upload.

```
from obs import ObsClient
from obs import PutObjectHeader
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the text to be uploaded.
    content = 'Hello OBS'
    # Specify the additional headers for object upload.
    headers = PutObjectHeader()
    # Calculate the CRC64 value of the data to be uploaded and submit the CRC64 value to the server for
    verification.
    headers.isAttachCrc64 = True
    # Upload the text.
    resp = obsClient.putContent(bucketName, objectKey, content, headers=headers)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Put Content Succeeded')
        print('requestId:', resp.requestId)
        print('crc64:', resp.body.crc64)
    else:
        print('Put Content Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

Sample Code 2: Performing a File-Based Upload

This example uses CRC64 to verify file consistency during single file upload.

```
from obs import ObsClient
from obs import PutObjectHeader
```

```
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
# Before running the sample code, ensure that the environment variables AccessKeyID and
SecretAccessKey have been configured.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional headers for object upload.
    headers = PutObjectHeader()
    # (Optional) Specify the MIME type of the object to be uploaded.
    headers.contentType = 'text/plain'
    # Calculate the CRC64 value of the data to be uploaded and submit the CRC64 value to the server for
    verification.
    headers.isAttachCrc64 = True
    bucketName = "examplebucket"
    # Specify the object name (the name after the file is uploaded).
    objectKey = "objectname"
    # Specify the full path of the file to be uploaded, for example, aa/bb.txt.
    file_path = 'localfile'
    # Specify the custom metadata of the file.
    metadata = {'meta1': 'value1', 'meta2': 'value2'}
    # Upload the file.
    resp = obsClient.putFile(bucketName, objectKey, file_path, metadata, headers)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Put File Succeeded')
        print('requestId:', resp.requestId)
        print('crc64:', resp.body.crc64)
        print('versionId:', resp.body.versionId)
        print('storageClass:', resp.body.storageClass)
    else:
        print('Put File Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Put File Failed')
    print(traceback.format_exc())
```

Sample Code 3: Performing an Append Upload

This example uses CRC64 to verify file consistency during append upload.

```
from obs import ObsClient
from obs import AppendObjectContent, AppendObjectHeader
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
```

```
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    headers = AppendObjectHeader()
    # Calculate the CRC64 value of the data to be uploaded and submit the CRC64 value to the server for
    verification.
    headers.isAttachCrc64 = True
    # Specify the message body for append upload.
    content = AppendObjectContent()
    # Specify the content to be appended.
    content.content = 'Hello OBS'
    # Specify the starting position (byte 0 in this example) the content is appended to.
    content.position = 0
    # If you upload an object for the first time using the append upload, an error will be reported (status
    code 409) if an ordinary object with the same name already exists.
    bucketName = "examplebucket"
    # Specify the name of the object to append content to.
    objectKey = "objectname"
    # Append content to the object.
    resp = obsClient.appendObject(bucketName, objectKey, content, headers=headers)

    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Append Object Succeeded')
        print('requestId:', resp.requestId)
        print('crc64:', resp.body.crc64)
        print('nextPosition:', resp.body.nextPosition)
    else:
        print('Append Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Append Object Failed')
    print(traceback.format_exc())
```

Sample Code 4: Performing a Resumable Upload

This example uses CRC64 to verify file consistency during resumable upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)

try:
    bucketName = "examplebucket"
    # Specify the object name (the name after the file is uploaded).
```

```
objectKey = "objectname"
# Specify the path of the file to be uploaded.
uploadFile = 'localfile'
# Specify the number of parts that can be concurrently uploaded.
taskNum = 5
# Specify the part size, in bytes. 10 MB is used as an example.
partSize = 10 * 1024 * 1024
# Enable the resumable upload by setting enableCheckpoint to True.
enableCheckpoint = True
# Upload the object using resumable upload.
resp = obsClient.uploadFile(bucketName, objectKey, uploadFile, partSize, taskNum, enableCheckpoint,
encoding_type='url', isAttachCrc64=True)

# If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
if resp.status < 300:
    print('Upload File Succeeded')
    print('requestId:', resp.requestId)
    print('crc64:', resp.body.crc64)
else:
    print('Upload File Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Upload File Failed')
    print(traceback.format_exc())
```

Code Example 5: Performing a Multipart Upload

This example uses CRC64 to verify file consistency during multipart upload.

```
# -*- coding:utf-8 -*-
from obs import ObsClient, CompleteMultipartUploadRequest, CompletePart
import os
import traceback
# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    #Specify the bucket name.
    bucketName = "examplebucket"
    #Specify the object name.
    objectKey = "objectname"
    # Specify the MIME type of the object.
    contentType = 'text/plain'
    # Initiate a multipart upload.
    resp = obsClient.initiateMultipartUpload(bucketName, objectKey,
                                             contentType=contentType)

    #Obtain the uploadId.
    uploadId = resp.body["uploadId"]
    #Specify the size of the part to upload.
    partSize = 10 * 1024 * 1024
    #Specify the part number.
    partNum = 1
    # Specify whether object indicates the file path. True is used here. The default value is False.
    isFile = True
    #Specify the local object file to upload.
```

```
filepath = r"D:\tmp\file.txt"
contentLength = os.path.getsize(filepath)
#Specify the start offset of a part in the source file.
offset = 0
etags = {}
crc64s = {}

while offset < contentLength:
    partSize = min(partSize, (contentLength - offset));
    # Upload parts.
    resp1 = obsClient.uploadPart(bucketName, objectKey, partNum, uploadId, filepath, isFile, partSize,
offset, isAttachCrc64=True)
    etags[partNum] = resp1.body.etag
    crc64s[partNum] = resp1.body.crc64
    offset = offset + partSize
    partNum = partNum + 1

completes = []
for i in range(1, partNum):
    completes.append(CompletePart(i, etags[i], crc64s[i], partSize))
# Assemble parts.
completeMultipartUploadRequest = CompleteMultipartUploadRequest(parts = completes)
resp = obsClient.completeMultipartUpload(bucketName, objectKey, uploadId,
completeMultipartUploadRequest, isAttachCrc64=True)
# If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
if resp.status < 300:
    print('Upload Part Succeeded')
    print('requestId:', resp.requestId)
    print('crc64:', resp.body.crc64)
else:
    print('Upload Part Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('multipartUpload Failed')
    print(traceback.format_exc())
```

Sample Code 6: Performing a File-Based Download

This example uses CRC64 to verify file consistency during file download.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the full path (localfile as an example) to which the file is downloaded. The full path contains
the local file name.
    downloadPath = 'localfile'
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Download the file.
    resp = obsClient.getObject(bucketName, objectKey, downloadPath, isAttachCrc64=True)
```



```
# If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
if resp.status < 300:
    print('Get Object Succeeded')
    print('requestId:', resp.requestId)
    print('crc64:', resp.body.crc64)
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

Sample Code 7: Performing a Resumable Download

This example uses CRC64 to verify file consistency during resumable download.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the full path to which the object is downloaded. The full path contains the local file name.
    downloadFile = 'localfile'
    # Specify the number of parts that can be concurrently downloaded.
    taskNum = 5
    # Specify the part size.
    partSize = 10 * 1024 * 1024
    # Enable the resumable download by setting enableCheckpoint to True.
    enableCheckpoint = True
    # Download the object using resumable download.
    resp = obsClient.downloadFile(bucketName, objectKey, downloadFile, partSize, taskNum,
    enableCheckpoint, isAttachCrc64=True)

    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Download File Succeeded')
        print('requestId:', resp.requestId)
        print('crc64:', resp.body.crc64)
    else:
        print('Download File Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Download File Failed')
    print(traceback.format_exc())
```

Sample Code 8: Performing a Binary Download

This example uses CRC64 to verify file consistency during binary download.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # If loadStreamInMemory is set to True, downloadpath will be invalid. The binary stream of the file
will be downloaded to the memory.
    #Download the object in a binary stream.
    resp = obsClient.getObject(bucketName=bucketName, objectKey=objectKey, loadStreamInMemory=True,
isAttachCrc64=True)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp.requestId)
        # Obtain the object content.
        print('buffer:', resp.body.buffer)
        print('size:', resp.body.size)
        print('crc64:', resp.body.crc64)
    else:
        print('Get Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

Sample Code 9: Performing a Streaming Download

This example uses CRC64 to verify file consistency during streaming download.

```
from obs import GetObjectRequest
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"
```

```
# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional parameters for object download.
    getObjectRequest = GetObjectRequest()
    # Rewrite the Content-Type header in the response.
    getObjectRequest.content_type = 'text/plain'
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Download the object using streaming.
    resp = obsClient.getObject(bucketName=bucketName, objectKey=objectKey,
    getObjectRequest=getObjectRequest, loadStreamInMemory=False, isAttachCrc64=True)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp.requestId)
        print('crc64 in obs:', resp.body.crc64)
        # Read the object content.
        while True:
            chunk = resp.body.response.read(65536)
            if not chunk:
                break
            print(chunk)
        resp.body.response.close()
    else:
        print('Get Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

Sample Code 10: Performing a Range-Based Download

This example uses CRC64 to verify file consistency during range download.

```
from obs import GetObjectHeader
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the additional headers for object download.
    headers = GetObjectHeader()
    # Specify the range (0-1000 in this example) of the object bytes to be returned.
    # You can specify multiple ranges when obtaining an object. For example,
headers.range='0-1000,2000-3000'.
    headers.range = '0-1000'
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Perform the range download. If loadStreamInMemory is set to True, downloadpath will be invalid,
```

```
and data streams will be downloaded to the memory.
resp = obsClient.getObject(bucketName, objectKey, loadStreamInMemory=True, headers=headers)
# If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
if resp.status < 300:
    print('Get Object Succeeded')
    print('requestId:', resp.requestId)
    # Obtain the object content.
    print('buffer:', resp.body.buffer)
    print('crc64:', resp.body.crc64)
else:
    print('Get Object Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

Sample Code 11: Obtaining Object Metadata

This example obtains the CRC64 value of **objectname** in bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables (recommended) or import it in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them using
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Obtain object metadata.
    resp = obsClient.getObjectMetadata(bucketName, objectKey)
    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Get Object Metadata Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
        print('crc64:', resp.body.crc64)
        print('lastModified:', resp.body.lastModified)
        print('contentType:', resp.body.contentType)
        print('contentLength:', resp.body.contentLength)
    else:
        print('Get Object Metadata Failed')
        print('requestId:', resp.requestId)
        print('status:', resp.status)
        print('reason:', resp.reason)
except:
    print('Get Object Metadata Failed')
    print(traceback.format_exc())
```

9 APIs Related to Multipart Upload (SDK for Python)

9.1 Multipart Upload Overview (SDK for Python)

You can upload large files using multipart upload. Multipart upload is applicable to many scenarios, including:

- Files to be uploaded are larger than 100 MB.
- The network condition is poor. Connection to the OBS server is constantly down.
- Sizes of files to be uploaded are uncertain.

A multipart upload consists of the following steps:

1. **Initiate a multipart upload** (`ObsClient.initiateMultipartUpload`).
2. **Upload parts one by one or concurrently** (`ObsClient.uploadPart`).
3. **Assemble parts** (`ObsClient.completeMultipartUpload`) or **abort the multipart upload** (`ObsClient.abortMultipartUpload`).

This example shows a complete multipart upload, including initiating a multipart upload, uploading parts, and assembling parts.

```
# -*- coding:utf-8 -*-
from obs import ObsClient, CompleteMultipartUploadRequest, CompletePart
import os
import traceback
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
```

```
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    #Bucket name
    bucketName = "examplebucket"
    #Object name
    objectKey = "objectname"
    # Specify the MIME type for the object.
    contentType = 'text/plain'
    # Initiate a multipart upload.
    resp = obsClient.initiateMultipartUpload(bucketName, objectKey,
                                             contentType=contentType)

    #Obtain the uploadId.
    uploadId = resp.body["uploadId"]
    #Specify the size of the part to upload.
    partSize = 512 * 1024 * 1024
    #Specify the part number.
    partNum = 1
    # Specify whether object indicates the file path. True is used here. The default value is False.
    isFile = True
    #Specify the local object file to upload.
    filepath = r"D:\tmp\file.txt"
    contentLength = os.path.getsize(filepath)
    #Start offset of a part in the source file.
    offset = 0
    etags = {}

    while offset < contentLength:
        partSize = min(partSize, (contentLength - offset));
        # Upload parts.
        resp1 = obsClient.uploadPart(bucketName, objectKey, partNum, uploadId, filepath, isFile, partSize,
                                     offset)
        etags[partNum] = resp1.body.etag
        offset = offset + partSize
        partNum = partNum + 1

    completes = []
    for i in range(1, partNum):
        completes.append(CompletePart(i, etags[i]))
    # Assemble parts.
    completeMultipartUploadRequest = CompleteMultipartUploadRequest(parts = completes)
    resp = obsClient.completeMultipartUpload(bucketName, objectKey, uploadId,
                                             completeMultipartUploadRequest)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Upload Part Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
    else:
        print('Upload Part Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('multipartUpload Failed')
    print(traceback.format_exc())
```

Below lists other multipart upload operations:

- [Listing Uploaded Parts \(SDK for Python\)](#)
- [Listing Multipart Uploads \(SDK for Python\)](#)
- [Copying a Part \(SDK for Python\)](#)
- [Aborting a Multipart Upload \(SDK for Python\)](#)

9.2 Initiating a Multipart Upload (SDK for Python)

Function

This API initiates a multipart upload and returns a globally unique upload ID. You can use the returned upload ID when you later upload, assemble, or list parts. A single object can be involved in multiple multipart uploads. Each multipart upload initiation request can contain additional headers such as **acl**, **metadata**, **contentType**, and **encoding_type**. These headers are recorded in the multipart upload metadata.

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).
- After initiating a multipart upload and uploading one or more parts, you must assemble the parts or abort the multipart upload. Only after that can OBS stop billing you for storing the uploaded parts.

Method

```
ObsClient.initiateMultipartUpload(bucketName, objectKey, acl, storageClass, metadata,  
websiteRedirectLocation, contentType, expires, extensionGrants, extensionHeaders)
```

Request Parameters

Table 9-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
acl	str	No	Explanation: ACL that can be pre-defined when a bucket is created. Value range: For details about the available options, see HeadPermission. Default value: PRIVATE
storageClass	str	No	Explanation: Object storage class. If this parameter is not set, the object inherits the storage class of its bucket. Value range: See Table 9-3. Default value: STANDARD

Parameter	Type	Mandatory (Yes/No)	Description
metadata	dict	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. To measure the custom metadata, sum the number of bytes in the UTF-8 encoding of each key and value. • 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

Parameter	Type	Mandator y (Yes/No)	Description
websiteRedirect- Location	str	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 http://www.example.com/: 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 for objects in the root directory of a bucket. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
contentType	str	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. Value range: See What Is Content-Type (MIME)? (Python SDK) 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.
expires	int	No	Explanation: Expiration time of the object (calculated from the latest modification time of the object). Expired objects are automatically deleted. Restrictions: This parameter can be configured only when uploading the object. It cannot be modified by calling a metadata modification API. Value range: An integer greater than or equal to 0, in days Default value: None
extensionGrants	list of Extension Grant	No	Explanation: List of the extended permissions for the object. For details, see Table 9-4 . Default value: None

Parameter	Type	Mandator y (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 9-2 HeadPermission

Constant	Default Value	Description
HeadPermission.PRIVATE	private	Private read/write A bucket or object can only be accessed by its owner.
HeadPermission.PUBLIC_READ	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 it is granted on an object, anyone can read the content and metadata of the object.
HeadPermission.PUBLIC_READ_WRITE	public-read-write	Public read/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 it is granted on an object, anyone can read the content and metadata of the object.

Constant	Default Value	Description
HeadPermission.PUBLIC_READ_DELIVERED	public-read-delivered	Public read on a bucket as well as 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 PUBLIC_READ_DELIVERED cannot be applied to objects.
HeadPermission.PUBLIC_READ_WRITE_DELIVERED	public-read-write-delivered	Public read/write on a bucket as well as 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 PUBLIC_READ_WRITE_DELIVERED cannot be applied to objects.
HeadPermission.BUCKET_OWNER_FULL_CONTROL	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 owned by another user, the bucket owner does not have the permissions on your object. After you grant this permission to the bucket owner, the bucket owner can have full control over your object.

Table 9-3 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.

Parameter	Type	Description
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 9-4 ExtensionGrant

Parameter	Type	Mandatory (Yes/No)	Description
granteeId	str	No	Explanation: Account (domain) ID of the grantee Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None
permission	str	No	Explanation: Granted permissions. For details, see Table 9-5 . Default value: None

Table 9-5 Permission

Constant	Description
READ	Read permission 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.
WRITE	Write permission A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. Such permission for an object is not applicable.
READ_ACP	Permission to read ACL configurations A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently.
WRITE_ACP	Permission to modify ACL configurations A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission for the bucket or object permanently. A grantee with this permission can modify the ACL, thus obtaining full access permissions.
FULL_CONTROL	Full control access, including read and write permissions for a bucket and its ACL, or for an object and its ACL. A grantee with this permission for a bucket has READ , WRITE , READ_ACP , and WRITE_ACP permissions for the bucket. A grantee with this permission for an object has READ , READ_ACP , and WRITE_ACP permissions for the object.

Responses

Table 9-6 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-7 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-8 GetResult.body

GetResult.body Type	Description
InitiateMultipartUploadResponse	Explanation: Response to the request for initiating a multipart upload. For details, see Table 9-9 .

Table 9-9 InitiateMultipartUploadResponse

Parameter	Type	Description
bucketName	str	Explanation: Name of the bucket involved in the multipart upload 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
objectKey	str	Explanation: Name of the object to be uploaded. 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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
uploadId	str	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 1 to 32 characters. Default value: None

Code Examples

This example initiates a multipart upload for bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify a pre-defined ACL (PRIVATE as an example).
    acl = 'PRIVATE'
    # Specify a storage class (STANDARD as an example) for the object.
    storageClass = 'STANDARD'
    # Specify a custom metadata of the object.
    metadata = {'key': 'value'}
    # Specify the MIME type for the object.
    contentType = 'text/plain'
    # Specify the lifecycle (7 as an example) for the object, in days.
    expires = 7
    # Initiate a multipart upload.
    resp = obsClient.initiateMultipartUpload(bucketName, objectKey, acl, storageClass, metadata,
                                             contentType=contentType, expires=expires, encoding_type='url')

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Initiate Multipart Upload Succeeded')
        print('requestId:', resp.requestId)
        print('bucketName:', resp.body.bucketName)
        print('objectKey:', resp.body.objectKey)
        print('uploadId:', resp.body.uploadId)
    else:
        print('Initiate Multipart Upload Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
```

```
print('errorMessage:', resp.errorMessage)
except:
    print('Initiate Multipart Upload Failed')
    print(traceback.format_exc())
```

9.3 Uploading a Part (SDK for Python)

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. Except for the part lastly being uploaded whose size ranges from 0 to 5 GB, sizes of the other parts range from 100 KB to 5 GB. Part numbers can be any number from 1 to 10,000.

When uploading a part, you must specify its upload ID and part number. A part number uniquely identifies a part and its position in the object you are uploading. If you upload a new part with the same part number as that of a previous part, the previously uploaded part will be overwritten. Whenever you upload a part, OBS returns the ETag header in the response. For each part upload task, you must record the part number and ETag value. These values are required in subsequent requests for you to complete a multipart upload.

Restrictions

- To upload a part, you must be the bucket owner or have the required permission (**obs:object:PutObject** in IAM or **PutObject** in a bucket policy).
- After initiating a multipart upload and uploading one or more parts, you must assemble the parts or abort the multipart upload. Only after that can OBS stop billing you for storing the uploaded parts.
- **partNumber** in a multipart upload must be unique. When the same **partNumber** of the same object is concurrently uploaded, last write wins policy is applied. The time of last write is defined as the time when the part metadata is created. To ensure data accuracy, the client must be locked to ensure concurrent uploads of the same part of the same object. Concurrent uploads for different parts of the same object do not need to be locked.

Method

```
ObsClient.uploadPart(bucketName, objectKey, partNumber, uploadId, object, isFile, partSize, offset, isAttachMd5, md5, progressCallback, autoClose, extensionHeaders)
```

Request Parameters

Table 9-10 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://<i>Bucket name.Domain name/Folder directory level/Object name</i>. If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
partNumber	int	Yes	Explanation: Part number Value range: [1,10000] Default value: None
uploadId	str	Yes	Explanation: Multipart upload ID which can be returned by initiating a multipart upload, for example, 000001648453845DBB78F2340DD460D8 Restrictions: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
object	str or readable object	Yes	Explanation: Part content to be uploaded Value range: A string or a readable object NOTE If object is a readable object that contains the read attribute, data is read from the readable object. Otherwise, the object content is a string. Default value: None
isFile	bool	No	Explanation: Whether object indicates the file path. Value range: True: object indicates the file path. False: object does not indicate the file path. Default value: False
offset	int	No	Explanation: Start offset of a part in the source file Value range: A non-negative integer not exceeding the size of the object to be uploaded, in bytes Default value: 0

Parameter	Type	Mandatory (Yes/No)	Description
partSize	int	No	Explanation: Part size Restrictions: • Except for the last uploaded part, all parts must be larger than 100 KB. 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. • 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. Value range: The value ranges from 100 KB to 5 GB, in bytes. Default value: 102400
isAttachMd5	bool	No	Explanation: Whether to automatically calculate the MD5 value of the data to be uploaded. Restrictions: If isAttachMd5 and md5 are used at the same time, isAttachMd5 is invalid. Value range: True: The MD5 value of the data to be uploaded is automatically calculated. False: The MD5 value of the data to be uploaded is not automatically calculated. Default value: False

Parameter	Type	Mandatory (Yes/No)	Description
md5	str	No	Explanation: Base64-encoded MD5 value of the part to be uploaded, which uniquely identifies the content of the uploaded part and can be used to identify whether the object content is changed. Restrictions: If isAttachMd5 and md5 are used at the same time, isAttachMd5 is invalid. Value range: The value must contain 32 characters. Default value: None
progressCallback	callable	No	Explanation: Callback function for obtaining the upload progress Default value: None NOTE This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (in seconds).
autoClose	bool	No	Explanation: Whether to automatically close data streams after the upload is complete Value range: True: The data stream is automatically closed. False: The data stream is not automatically closed. Default value: True

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 9-11 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-12 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None

Parameter	Type	Description
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None

Parameter	Type	Description
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-13 GetResult.body

GetResult.body Type	Description
UploadPartResponse	Explanation: Response to the request for uploading a part

Table 9-14 UploadPartResponse

Parameter	Type	Description
etag	str	Explanation: Base64-encoded 128-bit MD5 digest of a part. ETag is the unique identifier of the part content. It can be used to determine whether the part content is changed. Value range: The value must contain 32 characters. Default value: None

Code Examples

This example uploads a part.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
```

```
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Specify the name of the object to be uploaded to the bucket.
    objectKey = "objectname"
    # Specify the part number, which ranges from 1 to 10,000
    partNumber = "your partNumber"
    # Specify the ID of the multipart upload.
    uploadId = "your uploadid"
    # Specify the content of the part to be uploaded as a string or readable object.
    object = 'Hello OBS'
    # Specify whether object indicates the file path. The default value is False.
    isFile = False
    # Specify the start offset (in bytes) of a part in the source file. The default value is 0.
    offset = 0
    # Specify the size (in bytes) of a part in the source file. The default value is the file size minus offset.
    partSize = 9 * 1024 * 1024
    # Specify whether to automatically calculate the MD5 value of the data to be uploaded. The default
    value is False.
    isAttachMd5 = True
    # Upload the part to a specified bucket using the multipart upload ID.
    resp = obsClient.uploadPart(bucketName, objectKey, partNumber, uploadId, object, isFile, partSize,
                                offset, isAttachMd5=isAttachMd5)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Upload Part Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
    else:
        print('Upload Part Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Upload Part Failed')
    print(traceback.format_exc())
```

9.4 Listing Uploaded Parts (SDK for Python)

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. A maximum of 1,000 uploaded parts can be returned in a response. If your multipart upload has more than 1,000 parts, you need to send multiple requests 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).

- A returned list can only be used for verification. After a multipart upload is complete, the result in the list is no longer valid. However, when part numbers and the ETag values returned by OBS are uploaded, the list of part numbers specified by the user will be reserved.

Method

`ObsClient.listParts(bucketName, objectKey, uploadId, maxParts, partNumberMarker, extensionHeaders)`

Request Parameters

Table 9-15 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://Bucket name.Domain name/Folder directory level/Object name. If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
uploadId	str	Yes	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 1 to 32 characters. Default value: None
maxParts	int	No	Explanation: Maximum number of uploaded 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

Parameter	Type	Mandatory (Yes/No)	Description
partNumberMarker	int	No	Explanation: Part number after which uploaded part listing begins Restrictions: OBS lists only parts with greater numbers than that specified by this parameter. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 9-16 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-17 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None

Parameter	Type	Description
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-18 GetResult.body

GetResult.body Type	Description
ListPartsResponse	Explanation: Response to the request for listing uploaded parts

Table 9-19 ListPartsResponse

Parameter	Type	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
uploadId	str	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 32 characters. Default value: None
initiator	Initiator	Explanation: Initiator of the multipart upload. For details, see Table 9-21 . Default value: None
owner	Owner	Explanation: Owner of the multipart upload, which is consistent with initiator . For details, see Table 9-22 . Default value: None
storageClass	str	Explanation: Object storage class Value range: See Table 9-20 . Default value: None
partNumberMarker	int	Explanation: Part number after which part listing begins, which is consistent with that set in the request Value range: An integer greater than or equal to 0 Default value: None

Parameter	Type	Description
nextPartNumberMarker	int	Explanation: Part number to start with for the next part listing request. nextPartNumberMarker is returned when not all the parts are listed. You can set partNumberMarker to this value in the next request to list the remaining parts. Value range: An integer greater than or equal to 0 Default value: None
maxParts	int	Explanation: Maximum number of parts that can be listed per page. This parameter is consistent with that set in the request. 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
isTruncated	bool	Explanation: Whether all results are returned in the response Value range: • true: Not all results are returned. • false: All results are returned. Default value: None
parts	list of Part	Explanation: List of uploaded parts. For details, see Table 9-23. Default value: None

Table 9-20 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIERING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Table 9-21 Initiator

Parameter	Type	Mandatory (Yes/No)	Description
id	str	Yes if used as a request parameter	Explanation: Account (domain) ID of the initiator Value range: Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
name	str	No if used as a request parameter	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 9-22 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 9-23 Part

Parameter	Type	Description
partNumber	int	Explanation: Part number Value range: An integer ranging from 1 to 10000 Default value: None
lastModified	str	Explanation: Time when the part was last modified Default value: None
etag	str	Explanation: Part ETag, which is a Base64-encoded, 128-bit MD5 value Value range: The value must contain 32 characters. Default value: None
size	int	Explanation: Part size Default value: None

Code Examples

This example lists the parts that have been uploaded in a multipart upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
```

```
bucketName = "examplebucket"
objectKey = "objectname"
# Specify the ID of the multipart upload.
uploadId = "your uploadid"
# Specify the maximum number (10 as an example) of parts that can be listed per page.
maxParts = 10
# List the uploaded parts.
resp = obsClient.listParts(bucketName, objectKey, uploadId, maxParts, encoding_type='url')

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('List Parts Succeeded')
    print('requestId:', resp.requestId)
    print('bucketName:', resp.body.bucketName)
    print('objectKey:', resp.body.objectKey)
    print('uploadId:', resp.body.uploadId)
    print('storageClass:', resp.body.storageClass)
    print('isTruncated:', resp.body.isTruncated)
    print('initiator:', resp.body.initiator)
    print('owner:', resp.body.owner)

    index = 1
    for part in resp.body.parts:
        print('part [' + str(index) + ']')
        print('partNumber:', part.partNumber)
        print('lastModified:', part.lastModified)
        print('etag:', part.etag)
        print('size:', part.size)
        index += 1
else:
    print('List Parts Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('List Parts Failed')
    print(traceback.format_exc())
```

9.5 Listing Multipart Uploads (SDK for Python)

Function

This API lists ongoing multipart uploads.

Ongoing multipart uploads are the multipart uploads that have been initiated but have not been completed or aborted. A maximum of 1,000 multipart uploads can be returned in a response. If there are over 1,000 ongoing tasks satisfying the list criteria, you need to send more requests to query additional multipart uploads.

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).
- To list ongoing multipart uploads in a bucket, you must have the **ListBucketMultipartUploads** permission.
- By default, only the bucket owner and multipart upload initiator have this permission. The bucket owner can grant others the **ListBucketMultipartUploads** permission.

Method

```
ObsClient.listMultipartUploads(bucketName, multipart, extensionHeaders)
```

Request Parameters

Table 9-24 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucket	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
multipart	ListMultipartUploadsRequest	No	Explanation: Request parameters for listing multipart uploads. For details, see Table 9-25 .

Parameter	Type	Mandatory (Yes/No)	Description
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Table 9-25 ListMultipartUploadsRequest

Parameter	Type	Mandatory (Yes/No)	Description
prefix	str	No	Explanation: Name prefix that the objects 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 of logs/day1 , logs/day2 , and logs/day3 will be returned. If you leave this parameter blank, all multipart uploads in the bucket will be returned. Restrictions: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
max_uploads	int or str	No	Explanation: Maximum number of listed multipart uploads 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
delimiter	str	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 CommonPrefix. If no prefix is specified, objects with the same string from the first character to the first delimiter are grouped into one CommonPrefix. 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 CommonPrefix with abcd as the prefix. If only delimiter is set to d, objects abcd and abcde are grouped into a CommonPrefix with abcd as the prefix, and bbcde is grouped separately into another CommonPrefix with bbcd as the prefix. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
key_marker	str	No	Explanation: Object name after which the multipart upload listing begins Value range: The value of nextKeyMarker in the response body of the last request Default value: None
upload_id_marker	str	No	Explanation: Upload ID after which the multipart upload listing begins Restrictions: This parameter is valid only when used with key_marker . 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

Responses

Table 9-26 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-27 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-28 GetResult.body

GetResult.body Type	Description
ListMultipartUploadsResponse	Explanation: Response to the request for listing multipart uploads

Table 9-29 ListMultipartUploadsResponse

Parameter	Type	Description
bucket	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
keyMarker	str	Explanation: Object name after which the multipart upload listing begins, which is consistent with that set in the request. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
uploadIdMarker	str	Explanation: Upload ID after which the multipart upload listing begins, which is consistent with that set in the request Value range: The value must contain 1 to 32 characters. Default value: None
nextKeyMarker	str	Explanation: Object name to start with for the next multipart upload listing request. nextKeyMarker is returned when not all the objects are listed. You can set key_marker to this value in the next request to list the remaining multipart uploads. Value range: The value must contain 1 to 1,024 characters. Default value: None
nextUploadIdMarker	str	Explanation: Upload ID to start with for the next multipart upload listing request. This parameter is used together with nextKeyMarker. nextUploadIdMarker is returned when not all the objects are listed. You can set upload_id_marker to this value in the next request to list the remaining multipart uploads. Value range: The value must contain 1 to 32 characters. Default value: None

Parameter	Type	Description
maxUploads	int	Explanation: Maximum number of listed multipart uploads, which is consistent with that in the request 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
isTruncated	bool	Explanation: Whether all results are returned in the response Value range: • true: Not all results are returned. • false: All results are returned. Default value: None
prefix	str	Explanation: Prefix that the object names in the multipart uploads to be listed must contain. This parameter 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, the multipart uploads of logs/day1, logs/day2, and logs/day3 will be returned. If you leave this parameter blank, all multipart uploads in the bucket will be returned. Restrictions: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Description
delimiter	str	Explanation: A character used to group object names in multipart uploads, which is consistent with that set 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 grouped separately into another CommonPrefixes with bbcd as the prefix. Value range: The value must contain 1 to 1,024 characters. Default value: None
upload	list of Upload	Explanation: List of multipart uploads. For details, see Table 9-30. Default value: None
commonPrefixes	list of CommonPrefix	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 9-30 Upload

Parameter	Type	Description
key	str	Explanation: Name of the object to be uploaded. 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. Value range: The value must contain 1 to 1,024 characters. Default value: None
uploadId	str	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 1 to 32 characters. Default value: None
initiator	Initiator	Explanation: Initiator of the multipart upload. For details, see Table 9-32. Default value: None
owner	Owner	Explanation: Owner of the multipart upload, which is consistent with initiator. For details, see Table 9-33. Default value: None
storageClass	str	Explanation: Storage class of the object to be uploaded Value range: See Table 9-34. Default value: None

Parameter	Type	Description
initiated	str	Explanation: Time when the multipart upload is initiated Restrictions: The time must be in the ISO8601 format, for example, 2018-01-01T00:00:00.000Z . Sample code: DateTime(year=2023, month=9, day=12) Default value: None

Table 9-31 CommonPrefix

Parameter	Type	Description
prefix	str	Explanation: List of object name prefixes grouped according to the delimiter parameter (if specified) Value range: The value must contain 1 to 1,024 characters. Default value: None

Table 9-32 Initiator

Parameter	Type	Mandatory (Yes/No)	Description
id	str	Yes if used as a request parameter	Explanation: Account (domain) ID of the initiator Value range: Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
name	str	No if used as a request parameter	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 9-33 Owner

Parameter	Type	Mandatory (Yes/No)	Description
owner_id	str	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 IAM User ID? (SDK for Python) Default value: None
owner_name	str	No if used as a request parameter	Explanation: Account name of the owner Value range: To obtain the account ID, see How Do I Get My Account ID and IAM User ID? (SDK for Python) Default value: None

Table 9-34 StorageClass

Parameter	Type	Description
STANDARD	Standard storage class	Explanation: 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.
WARM	Warm storage class	Explanation: Used for storing data that is semi-frequently accessed (fewer than 12 times a year) but is instantly available when needed.
COLD	Cold storage class	Explanation: Used for storing rarely accessed (once a year) data.
INTELLIGENT_TIER RING	Intelligent Tiering	Explanation: Is designed to optimize storage costs by automatically moving data to a more economical access tier when data access patterns change. This storage class is ideal for data with constantly changing or unpredictable access patterns.

Code Examples

This example lists multipart uploads in a bucket.

```
from obs import ObsClient
from obs import ListMultipartUploadsRequest
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Specify the prefix that the object names in the multipart uploads to be listed must contain.
    prefix = 'prefix'
    # Specify the maximum number (10 as an example) of returned multipart uploads. The value ranges
```



```
from 1 to 1,000. If the value is not in this range, 1,000 is returned by default.
max_uploads = 10
# Specify request parameters for listing multipart uploads.
multipart = ListMultipartUploadsRequest(prefix=prefix, max_uploads=max_uploads)
# List multipart uploads in a bucket.
resp = obsClient.listMultipartUploads(bucketName, multipart, encoding_type='url')

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('List Multipart Uploads Succeeded')
    print('requestId:', resp.requestId)
    print('bucket:', resp.body.bucket)
    print('prefix:', resp.body.prefix)
    print('maxUploads:', resp.body.maxUploads)
    print('isTruncated:', resp.body.isTruncated)
    index = 1
    for upload in resp.body.upload:
        print('upload [' + str(index) + ']')
        print('key:', upload.key)
        print('uploadId:', upload.uploadId)
        print('storageClass:', upload.storageClass)
        print('initiated:', upload.initiated)
        print('owner_id:', upload.owner.owner_id)
        print('owner_name:', upload.owner.owner_name)
        index += 1
    else:
        print('List Multipart Uploads Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('List Multipart Uploads Failed')
    print(traceback.format_exc())
```

9.6 Assembling Parts (SDK for Python)

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 the same way the information in a common object upload is processed. In the case of assembling parts concurrently, last write wins is applied, 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 creates an object by putting part numbers in ascending order. If any object metadata is provided in the initiation of the multipart upload, OBS will associate the metadata with the object. After the multipart upload is complete, the parts will no longer exist. A part assembling request must contain the upload ID, part numbers, and a list of corresponding ETag values. In response to the request, the ETag that uniquely identifies the assembled parts is contained. This ETag is not the MD5 hash value of the entire object.

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).
- After a multipart upload is complete, the uploaded parts that are not assembled will be automatically deleted and cannot be recovered. Before assembling parts, use the API for listing uploaded parts to check all parts to ensure that no part is missed.

Method

```
ObsClient.completeMultipartUpload(bucketName, objectKey, uploadId, completeMultipartUploadRequest, extensionHeaders)
```

Request Parameters

Table 9-35 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://Bucket name.Domain name/Folder directory level/Object name. If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
uploadId	str	Yes	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 32 characters. Default value: None
completeMultipartUploadRequest	CompleteMultipartUploadRequest	Yes	Explanation: Request parameters for assembling parts. For details, see Table 9-36.
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python). Default value: None

Table 9-36 CompleteMultipartUploadRequest

Parameter	Type	Mandatory (Yes/No)	Description
parts	list of CompletePart	Yes	Explanation: List of parts to be assembled. For details, see Table 9-37 .

Table 9-37 CompletePart

Parameter	Type	Mandatory (Yes/No)	Description
partNum	int	Yes	Explanation: Part number Value range: [1,10000] Default value: None
etag	str	Yes	Explanation: ETag of a part. It is calculated by encoding the 128-bit MD5 value of the part using Base64. Value range: The value must contain 32 characters. Default value: None

Responses

Table 9-38 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-39 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-40 GetResult.body

GetResult.body Type	Description
CompleteMultipartUploadResponse	Explanation: Response to the request for assembling parts. For details, see Table 9-41 .

Table 9-41 CompleteMultipartUploadResponse

Parameter	Type	Description
etag	str	Explanation: The ETag that uniquely identifies the object after its parts were assembled, calculated based on the ETag of each part. 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
bucket	str	Explanation: Bucket in which parts are assembled 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Description
key	str	Explanation: Object name obtained after part assembling. 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. Value range: The value must contain 1 to 1,024 characters. Default value: None
location	str	Explanation: URL of the generated object after part assembling Default value: None
versionId	str	Explanation: Version ID of the object obtained after part assembling Value range: The value must contain 32 characters. Default value: None
objectUrl	str	Explanation: Full path to the obtained object after part assembling Default value: None

Code Examples

This example assembles the parts previously uploaded to a bucket.

```
from obs import ObsClient
from obs import CompleteMultipartUploadRequest, CompletePart
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
```

```
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify the list of parts to be assembled by configuring completeMultipartUploadRequest. Each part
    contains partNum and etag.
    part1 = CompletePart(partNum=1, etag='etag1')
    part2 = CompletePart(partNum=2, etag='etag2')
    completeMultipartUploadRequest = CompleteMultipartUploadRequest(parts=[part1, part2])

    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the ID of the multipart upload.
    uploadId = "your uploadid"
    # Assemble the parts uploaded to the bucket.
    resp = obsClient.completeMultipartUpload(bucketName, objectKey, uploadId,
    completeMultipartUploadRequest, encoding_type='url')

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Complete Multipart Upload Succeeded')
        print('requestId:', resp.requestId)
        print('etag:', resp.body.etag)
        print('bucket:', resp.body.bucket)
        print('key:', resp.body.key)
        print('location:', resp.body.location)
        print('versionId:', resp.body.versionId)
    else:
        print('Complete Multipart Upload Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Complete Multipart Upload Failed')
    print(traceback.format_exc())
```

9.7 Copying a Part (SDK for Python)

Function

This API copies a part to a specified bucket. This request must contain the multipart upload ID.

After creating a multipart upload, you can specify the upload ID and upload parts for the multipart upload. Alternatively, you can call this API to copy part or all of an uploaded object as a part.

Assume that you copy a source object and save it as **part1**. If there is already a **part1** before the copy, the new **part1** will overwrite the existing **part1**. Then, only the new **part1** can be listed and the old **part1** is deleted. Therefore, when using this API, ensure that the target part does not exist or is no longer used, to avoid accidental deletion. The source object keeps unchanged in the entire 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).

- You cannot determine whether a part copy is successful only based on **status_code** in the HTTP header returned. A **200** response indicates that the server has received the request and starts to process the request. Only when the response body contains ETag, the copy is successful, or the copy fails.

Method

```
ObsClient.copyPart(bucketName, objectKey, partNumber, uploadId, copySource, copySourceRange, extensionHeaders)
```

Request Parameters

Table 9-42 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	Yes	Explanation: Target 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. - If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://Bucket name.Domain name/Folder directory level/Object name . If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
partNumber	int	Yes	Explanation: Part number Value range: [1,10000] Default value: None
uploadId	str	Yes	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
copySource	str	Yes	Explanation: Parameter used to specify the source bucket, source object, and source object version ID which can be left blank. It is in the format of <i>SourceBucketName/SourceObjectName?versionId=SourceObjectVersionId</i>. 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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. • 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
copySourceRange	str	No	Explanation: Copy source range. For example, 0-999 indicates the range is from byte 1 to byte 1,000. Value range: Value range: 0 to the object length minus 1. Format: <i>x-y</i> , indicating the range is from byte <i>x+1</i> to byte <i>y+1</i> . Restrictions: The upper limit 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. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 9-43 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-44 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Table 9-45 GetResult.body

GetResult.body Type	Description
CopyPartResponse	Explanation: Response to the request for copying a part. For details, see Table 9-46 .

Table 9-46 CopyPartResponse

Parameter	Type	Description
etag	str	Explanation: Base64-encoded 128-bit MD5 digest of a part. ETag is the unique identifier of the part content. It can be used to determine whether the part content is changed. Value range: The value must contain 32 characters. Default value: None
lastModified	str	Explanation: Time when the last modification was made to the object Restrictions: The time must be in the GMT format, for example, Wed, 25 Mar 2020 02:39:52 GMT . Default value: None

Code Examples

This example copies parts from one bucket to another.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the part number, which ranges from 1 to 10,000
    partNumber = "your partNumber"
    # Specify the ID of the multipart upload.
    uploadId = "your uploadId"
    # Specify the source bucket, source object, and source object version ID (optional). The format is source-
    bucket-name/source-object-name?versionId=source-object-versionId.
```

```
copySource = 'sourcebucketname/sourceobjectkey'
# [Optional] Specify the copy scope in the format of x-y. Value range: [0, source-object-length - 1].
# copySourceRange = '0-15'
# Copy the part to the specified bucket using the multipart upload ID.
resp = obsClient.copyPart(bucketName, objectKey, partNumber, uploadId, copySource)

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Copy Part Succeeded')
    print('requestId:', resp.requestId)
    print('etagValue:', resp.body.etagValue)
    print('modifiedDate:', resp.body.modifiedDate)
else:
    print('Copy Part Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Copy Part Failed')
    print(traceback.format_exc())
```

9.8 Aborting a Multipart Upload (SDK for Python)

Function

This API aborts a multipart upload using the multipart upload ID.

After a multipart upload is aborted, the upload ID cannot be used to upload any part. The storage occupied by any uploaded parts will be released. If any part uploads are in progress, aborting the multipart upload might or might not make the uploads successful. To release the storage occupied by all uploaded parts, you can only 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).

Method

```
ObsClient.abortMultipartUpload(bucketName, objectKey, uploadId, extensionHeaders)
```

Request Parameters

Table 9-47 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://Bucket name.Domain name/Folder directory level/Object name . If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
uploadId	str	Yes	Explanation: Multipart upload ID, for example, 000001648453845DBB78F2340DD460D8 Value range: The value must contain 32 characters. Default value: None
extensionHeaders	dict	No	Explanation: Extension headers. Value range: See User-defined Headers (SDK for Python) . Default value: None

Responses

Table 9-48 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 9-49 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
requestId	str	Explanation: Request ID returned by the OBS server Default value: None

Parameter	Type	Description
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example aborts a multipart upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
```

```
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the ID of the multipart upload.
    uploadId = "your uploadid"
    # Abort the multipart upload.
    resp = obsClient.abortMultipartUpload(bucketName, objectKey, uploadId)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Abort Multipart Upload Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Abort Multipart Upload Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Abort Multipart Upload Failed')
    print(traceback.format_exc())
```

10 Client-Side Encryption APIs (SDK for Python)

10.1 Overview (SDK for Python)

Client-side encryption is a process where data is encrypted using the selected encryption method and information on your local PC before it is transmitted to an OBS server. During this process, the encryption method used and necessary information required for decryption will be stored in object metadata. During a download, the OBS SDK decrypts the data based on the key provided and the information stored in object metadata, and then returns the decrypted data.



WARNING

- OBS does not store your master key in any way. Keep your master key correct and intact. If your master key is lost or mistakenly used, your data cannot be decrypted.
- Do not modify the information required for decryption in object metadata when moving or replicating an encrypted object or modifying the object metadata. If you do so, your data cannot be decrypted.

Encryption Process and Cipher Suites

The OBS Python SDK offers two cipher suite generators: CTRCipherGenerator based on AES-CTR and CtrRSACipherGenerator based on RSA and AES-CTR.

If CTRCipherGenerator is used to upload objects, you need to provide a data key. The SDK then randomly generates an initial value for each object and uses the data key and initial value to encrypt the object. After the encryption, the SDK uploads the encrypted object to OBS and stores its initial value in object metadata. To download this object, you need to provide the corresponding data key. The SDK then automatically obtains the initial value from object metadata, uses the data key provided and initial value to decrypt the object, and returns the decrypted data. If a different data key is provided for download, the SDK returns an unavailable decrypted file.

If CtrRSACipherGenerator is used to upload objects, you need to provide an RSA public or private key. The SDK then randomly generates a data key and initial value for each object and uses the data key and initial value to encrypt the object. After that, the SDK uploads the encrypted object to OBS and then uses the RSA key to encrypt the data key. The encrypted data key and initial value are stored in the object metadata. To download this object, you need to provide the same RSA public or private key as you provided for the upload. The SDK then automatically obtains the data key and initial value stored in the object metadata and uses the provided key to decrypt the data key. If the provided key does not match the key used for the upload, an error will be reported. After the data key is decrypted, the SDK uses the decrypted data key and initial value to decrypt the object and returns the decrypted data.

API Changes

CryptoObsClient is inherited from ObsClient. Except the APIs listed in the following table, all other APIs of CryptoObsClient are the same as those of ObsClient.

Table 10-1 API description

API	ObsClient API Action	CryptoObsClient API Action
appendObject	Append data to an object.	Report an error.
copyPart	Copy an object part.	Report an error.
initiateMultipartUpload	Initialize a multipart upload.	Report an error.
uploadPart	Upload object parts.	Report an error.
putContent	Upload streams or texts.	Encrypt streams or texts and then upload them to OBS.
putFile	Upload common files to OBS.	Encrypt common files and then upload them to OBS.
getObject	Download files.	Decrypt files and then return the decrypted data.
uploadFile	Upload files in resumable mode to OBS.	Encrypt files and then upload them in resumable mode to OBS.
downloadFile	Download files in resumable mode to your local PC.	Decrypt files and then download them in resumable mode to your local PC.

API	ObsClient API Action	CryptoObsClient API Action
initiateEncryptedMulti-partUpload	None	Initialize an encrypted multipart upload.
uploadEncryptedPart	None	Encrypt uploaded parts.

Decryption Information in Metadata

The SDK saves the information required for decryption to the custom metadata of an object and does not back it up. If you modify the stored information, data cannot be decrypted. The table below describes the information required for decryption.

Table 10-2 Parameters

Parameter	Mandatory (Yes/No)	Description
encrypted-algorithm	Yes	Information about a cipher suite
encrypted-object-key	Yes (when CtrRSACipherGenerator is used)	Data key encrypted using an RSA key
encrypted-start	Yes	String initially encoded using Base64. It is used to encrypt an object.
master-key-info	No	Information about an encryption key
plaintext-sha256	No	SHA-256 value of an object before encryption
plaintext-content-length	No	Object length before encryption
encrypted-sha256	No	SHA-256 of an encrypted object

10.2 Client-Side Encryption APIs (SDK for Python)

Initializing CryptoCipher

The OBS SDK for Python offers two types of CryptoCipher. Choose the one right for you.

CtrRSACipherGenerator needs an RSA public or private key to encrypt the randomly generated data key.

Method

```
CtrRSACipherGenerator(master_crypto_key_path, master_key_info=None, need_sha256=False)
```

API Parameters**Table 10-3** CtrRSACipherGenerator

Parameter	Type	Mandatory (Yes/No)	Description
master_crypto_key_path	str	Yes	Explanation: Path where the RSA key file is stored Default value: None
master_key_info	str	No	Explanation: Key information, which is stored in the custom metadata of objects to help you distinguish different data keys. You need to maintain the mapping between master_key_info and crypto_key . Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
need_sha256	bool	No	Explanation: Whether to verify the SHA-256 value of the encrypted data and to add SHA-256 values calculated before and after the encryption to the custom metadata NOTE To reduce memory overheads, the SDK uses streaming computing, which means that a file needs to be read and encrypted twice in a common upload and three times in a resumable upload. Value range: True: The SHA-256 value of the encrypted data is verified. The SDK automatically calculates SHA-256 before and after the object is encrypted and saves the two values to the custom metadata of the object. In addition, the SDK adds the SHA-256 value of the encrypted object to the request header when sending a request to the server. After receiving the request, the server calculates the SHA-256 value of the object and checks the consistency between the calculated and the received values. If they are inconsistent, an error message is returned. False: The SHA-256 value of the encrypted data is not verified. Default value: False

CTRCipherGenerator requires only a data key. This key is used to encrypt all objects.

Method

```
CTRCipherGenerator(crypto_key, master_key_info=None, crypto_iv=None, need_sha256=False)
```

API Parameters

Table 10-4 CTRCipherGenerator

Parameter	Type	Mandatory (Yes/No)	Description
crypto_key	str or bytes	Yes	Explanation: Data key used for encrypting data Restrictions: - Length: 32 bytes - If a character string is passed, it is converted to bytes. Default value: None
master_key_info	str	No	Explanation: Key information, which is stored in the custom metadata of objects to help you distinguish different data keys. You need to maintain the mapping between master_key_info and crypto_key. Default value: None
crypto_iv	str or bytes	No	Explanation: Initial value used for encrypting data Restrictions: - Length: 16 bytes - If crypto_iv is specified, all objects are encrypted using the specified initial value. If crypto_iv is not specified, the SDK randomly generates an initial value for each object. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
need_sha256	bool	No	Explanation: Whether to verify the SHA-256 value of the encrypted data and to add SHA-256 values calculated before and after the encryption to the custom metadata NOTE To reduce memory overheads, the SDK uses streaming computing, which means that a file needs to be read and encrypted twice during a common upload, but in a resumable upload, a file needs to be read and encrypted three times. Value range: True: The SHA-256 value of the encrypted data is verified. The SDK automatically calculates SHA-256 before and after the object is encrypted and saves the two values to the custom metadata of the object. In addition, the SDK adds the SHA-256 value of the encrypted object to the request header when sending a request to the server. After receiving the request, the server calculates the SHA-256 value of the object and checks the consistency between the calculated and the received values. If they are inconsistent, an error message is returned. False: The SHA-256 value of the encrypted data is not verified. Default value: False

Initializing CryptoClient

Initializing CryptoClient is inherited from initializing an ObsClient. For details, see [Initializing an Instance of ObsClient \(SDK for Python\)](#).

Method

```
CryptoClient(access_key_id,secret_access_key,server,cipher_generator=your-cipher_generator, *args, **kwargs)
```

API Parameters

Table 10-5 List of request parameters

Parameter	Description	Recommended Value
access_key_id	Explanation: Access key ID (AK) Default value: An empty string, indicating an anonymous user	N/A
secret_access_key	Explanation: Secret access key (SK) Default value: An empty string, indicating an anonymous user	N/A
server	Explanation: Server address for accessing OBS. It consists of a protocol type, domain name, and port number, for example, https://your-endpoint:443 . For security purposes, you are advised to use HTTPS. Default value: None	N/A
cipher_generator	Explanation: Cipher generator used by a client Value range: • CtrRSACipherGenerator • CTRCipherGenerator Default value: None	N/A

Initiating an Encrypted Multipart Upload

Initiating an encrypted multipart upload is inherited from the ObsClient API for initiating a multipart upload. For details, see [Initiating a Multipart Upload \(SDK for Python\)](#).

Method

```
CryptoClient.initiateEncryptedMultipartUpload(bucketName, objectKey, crypto_cipher, *args, **kwargs)
```

API Parameters

Table 10-6 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://<i>Bucket name.Domain name/Folder directory level/Object name</i>. If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
crypto_cipher	OBSCipher	Yes	Explanation: OBS cipher generated by calling cipher_generator.new(). Configure this parameter into the metadata during task initiation. Value range: A value generated by calling cipher_generator.new(). cipher_generator is the cipher_generator parameter set for CryptoClient initialization. Example: cipher = ctr_client.cipher_generator.new(""). ctr_client is the encryption client. Default value: None

Uploading an Encrypted Part

Uploading encrypted parts is inherited from the ObsClient API for uploading parts. For details, see [Uploading a Part \(SDK for Python\)](#).

Method

```
CryptoClient.initiateEncryptedMultipartUpload(bucketName, objectKey, partNumber, uploadId,  
crypto_cipher, *args, **kwargs)
```

API Parameters

Table 10-7 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	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. Value range: The value must contain 1 to 1,024 characters. Default value: None NOTE The object URL is in the following format: https://Bucket name.Domain name/Folder directory level/Object name. If this object is stored in the root directory of the bucket, its URL does not contain the folder directory level.
partNumber	int	Yes	Explanation: Part number Value range: [1,10000] Default value: None
uploadId	str	Yes	Explanation: Multipart upload ID, which can be returned by initiating a multipart upload, for example, 000001648453845DBB78F2340DD460D8 Restrictions: The value must contain 32 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
crypto_cipher	OBSCipher	Yes	Explanation: OBS cipher generated by calling cipher_generator.new() Value range: A value generated by calling cipher_generator.new(). cipher_generator is the cipher_generator parameter set for CryptoClient initialization. Example: cipher = ctr_client.cipher_generator.new(""). ctr_client is the encryption client. Default value: None

11 Other APIs (SDK for Python)

11.1 Creating a Signed URL (SDK for Python)

Function

This API creates a URL whose **Query** parameters are carried with authentication information by specifying the AK and SK, HTTP method, and request parameters. You can provide other users with this URL for temporary access. When generating a URL, you need to specify the validity period of the URL to restrict the access duration of visitors.

If you want to grant other users the permission to perform other operations on buckets or objects (for example, upload or download objects), generate a URL with the corresponding request (for example, to upload an object using the URL that generates the PUT request) and provide the URL for other users.

Restrictions

- If a CORS or signature mismatch error occurs, refer to the following steps to troubleshoot the issue:
 - a. If CORS is not configured, you need to configure CORS rules on OBS Console.
 - b. If the signatures do not match, check whether signature parameters are correct. 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 is set to an incorrect value 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.

Method

```
ObsClient.createSignedUrl(method, bucketName, objectKey, specialParam, expires, headers, queryParams)
```

Request Parameters

Table 11-1 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
method	str	Yes	Explanation: HTTP methods Value range: • GET • POST • PUT • DELETE • HEAD Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	No	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None
objectKey	str	No	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. Value range: The value must contain 1 to 1,024 characters. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
specialParam	str	No	Explanation: Special operator, which indicates the sub-resource to be operated Value range: • versions • uploads • location • storageinfo • quota • storagePolicy • acl • append • logging • policy • lifecycle • website • versioning • cors • notification • delete • restore Default value: None
expires	int	No	Explanation: Expiration time of the signed URL 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	dict	No	Explanation: Headers in the request Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
queryParams	dict	No	Explanation: Query parameters in the request Default value: None

 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.
2. If the signatures do not match, check whether signature parameters are correct. 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 is set to an incorrect value 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.

Responses

Table 11-2 List of returned results

Parameter	Type	Description
signedUrl	str	Explanation: Signed URL Default value: None
actualSignedRequestHeaders	dict	Explanation: Actual headers in the request initiated by using the signed URL Default value: None

Code Examples

This example creates temporary signed URLs.

```
from obs import ObsClient
import os
import traceback
import base64
```

```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Create a signed URL for creating a bucket.
    res1 = obsClient.createSignedUrl(method='PUT', bucketName='bucketname', expires=3600)
    print('signedUrl:', res1.signedUrl)
    print('actualSignedRequestHeaders:', res1.actualSignedRequestHeaders)

    # Create a signed URL for uploading an object.
    res2 = obsClient.createSignedUrl(method='PUT', bucketName='bucketname', objectKey='objectkey',
expires=3600,
                                headers={'Content-Type': 'text/plain'})
    print('signedUrl:', res2.signedUrl)
    print('actualSignedRequestHeaders:', res2.actualSignedRequestHeaders)

    # Create a signed URL for setting an object ACL.
    res3 = obsClient.createSignedUrl(method='PUT', bucketName='bucketname', objectKey='objectkey',
specialParam='acl',
                                expires=3600, headers={'x-obs-acl': 'private'})
    print('signedUrl:', res3.signedUrl)
    print('actualSignedRequestHeaders:', res3.actualSignedRequestHeaders)

    # Create a signed URL for downloading an object.
    res4 = obsClient.createSignedUrl(method='GET', bucketName='bucketname', objectKey='objectkey',
expires=3600)
    print('signedUrl:', res4.signedUrl)
    print('actualSignedRequestHeaders:', res4.actualSignedRequestHeaders)

    # Create a signed URL for deleting an object.
    res5 = obsClient.createSignedUrl(method='DELETE', bucketName='bucketname', objectKey='objectkey',
expires=3600)
    print('signedUrl:', res5.signedUrl)
    print('actualSignedRequestHeaders:', res5.actualSignedRequestHeaders)

    # Create a signed URL for deleting a bucket.
    res6 = obsClient.createSignedUrl(method='DELETE', bucketName='bucketname', expires=3600)
    print('signedUrl:', res6.signedUrl)
    print('actualSignedRequestHeaders:', res6.actualSignedRequestHeaders)

    # Create a signed URL for initiating a multipart task.
    res7 = obsClient.createSignedUrl(method='POST', bucketName='bucketname', objectKey='objectkey',
specialParam='uploads', expires=3600)
    print('signedUrl:', res7.signedUrl)
    print('actualSignedRequestHeaders:', res7.actualSignedRequestHeaders)

    # Create a signed URL for uploading a part.
    res8 = obsClient.createSignedUrl(method='PUT', bucketName='bucketname', objectKey='objectkey',
expires=3600,
                                queryParams={'partNumber': '1', 'uploadId': '00000*****'})
    print('signedUrl:', res8.signedUrl)
    print('actualSignedRequestHeaders:', res8.actualSignedRequestHeaders)

    # Create a signed URL for assembling parts.
    res9 = obsClient.createSignedUrl(method='POST', bucketName='bucketname', objectKey='objectkey',
expires=3600,
```

```
        queryParams={'uploadId': '00000*****'})
    print('signedUrl:', res9.signedUrl)
    print('actualSignedRequestHeaders:', res9.actualSignedRequestHeaders)

    # Create a signed URL for image persistency.
    # Name of the bucket that stores the source object
    bucketName = 'originBucketName';
    # Source object name before the processing
    objectKey = 'test.png';

    # Name of the object after processing
    targetObjectName = "save.png"
    # (Optional) Name of the bucket that stores the new object
    targetBucketName = "saveBucketName"
    queryParams={}
    queryParams["x-image-process"]="image/resize,w_100"
    queryParams["x-image-save-object"]=base64.b64encode(targetObjectName.encode("utf-8")).decode()
    # Optional parameter
    queryParams["x-image-save-bucket"]=base64.b64encode(targetBucketName.encode("utf-8")).decode()

    res10 = obsClient.createSignedUrl(method='GET', bucketName=bucketName, objectKey=objectKey,
    queryParams=queryParams, expires=3600)
    print('signedUrl:', res10.signedUrl)
    print('actualSignedRequestHeaders:', res10.actualSignedRequestHeaders)
except:
    print(traceback.format_exc())
```

11.2 Generating Authentication Parameters for Browser-Based Uploads (SDK for Python)

Function

This API generates parameters for authentication. The parameters can be used to perform a browser-based upload through POST operations.

NOTE

There are two request parameters generated for authentication:

- **policy**, which corresponds to the **policy** parameter in the form.
- **signature**, which corresponds to the **signature** parameter in the form.

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).

Method

```
ObsClient.createPostSignature(bucketName, objectKey, expires, formParams)
```

Request Parameters

Table 11-3 List of request parameters

Parameter	Type	Mandatory (Yes/No)	Description
bucketName	str	No	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 periods (.) and hyphens (-) adjacent to each other, for example, my-.bucket or my.-bucket. • If you repeatedly create buckets of 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. Default value: None

Parameter	Type	Mandatory (Yes/No)	Description
objectKey	str	No	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. Value range: The value must contain 1 to 1,024 characters. Default value: None
expires	int	No	Explanation: Expiration time of authentication for a browser-based upload Value range: An integer greater than or equal to 0, in seconds Default value: 300
formParams	dict	No	Explanation: Parameters of browser-based uploads, not including key, policy, and signature. Value range: • acl • cache-control • content-type • content-disposition • content-encoding • expires Default value: None

Responses

Table 11-4 List of returned results

Parameter	Type	Description
originPolicy	str	Explanation: policy not encoded by Base64. This parameter can only be used for verification. Example: {"expiration":"2023-09-12T12:52:59Z", "conditions":[{"content-type":"text/plain"}, {"bucket":"examplebucket"}, {"key":"example/objectname"}],} Default value: None
policy	str	Explanation: Base64-encoded value of policy in the form Example: eyJleHBpcmF0aW9uljoiMjAyMy0wOS0xMlQxMjo1Mjo1OVoiLCJjb25kaXRpb25zljpbeyJjb250ZW50LXR5cGUiOiJ0ZXh0L3BsYWluIn0seyJidWNrZXQiOiJleGFtcGxlYnVja2V0In0seyJrZXkiOiJleGFtcGxlL29iamVjdG5hbWUifSxdfQ== Default value: None
signature	str	Explanation: signature in the form Example: g0jQr4v9VWd1Q2FOFDG6LGfV9Cw= Default value: None

Code Examples

This example generates authentication parameters **policy** and **signature** for a browser-based upload.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
```

```
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Configure the validity period (in seconds) for a browser-based upload request. 3600 is used as an
    example.
    expires = 3600
    # Specify parameters for a browser-based upload except key, policy, and signature. In this example, x-
    obs-acl is set to private and content-type is set to text/plain.
    formParams = {'x-obs-acl': 'private', 'content-type': 'text/plain'}
    # Create parameters for a browser-based upload.
    resp = obsClient.createPostSignature(bucketName, objectKey, expires, formParams)

    print('originPolicy:', resp.originPolicy)
    print('policy:', resp.policy)
    print('signature:', resp.signature)
except:
    print(traceback.format_exc())
```

11.3 Server-Side Encryption (SDK for Python)

Function

This API configures server-side encryption for objects, so that they will be encrypted or decrypted when you upload them to or download them from a bucket.

The encryption and decryption happen on the server side.

There are different encryption methods for you to choose from. Available encryption methods include server-side encryption with KMS-managed keys (SSE-KMS) and server-side encryption with customer-provided keys (SSE-C). Both of the two methods use the AES-256 algorithm.

With SSE-KMS, OBS uses the keys provided by KMS for server-side encryption.

With SSE-C, OBS uses the keys and MD5 values provided by customers for server-side encryption.

When server-side encryption is used, the returned ETag value is not the object's MD5 value. OBS will verify the object's MD5 value as long as the upload request includes the **Content-MD5** header, no matter whether server-side encryption is used or not.

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).

Method

```
ObsClient.putFile(bucketName, objectKey, file_path, metadata, headers, extensionHeaders)
```

Supported APIs

The following table lists APIs related to server-side encryption:

Method in OBS SDK for Python	Description	Supported Encryption Method
ObsClient.putContent	Sets the encryption algorithm and key during object upload to enable server-side encryption.	SSE-KMS SSE-C
ObsClient.putFile	Sets the encryption algorithm and key during file 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 task 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 multipart copy. 2. Sets the encryption algorithm and key during part copy to enable the encryption for the target part.	SSE-C

Responses

Table 11-5 List of returned results

Type	Description
GetResult	Explanation: SDK common results

Table 11-6 GetResult

Parameter	Type	Description
status	int	Explanation: HTTP status code 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). Default value: None
reason	str	Explanation: Reason description. Default value: None
errorCode	str	Explanation: Error code returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None
errorMessage	str	Explanation: Error message returned by the OBS server. If the value of status is less than 300 , this parameter is left blank. Default value: None

Parameter	Type	Description
requestId	str	Explanation: Request ID returned by the OBS server Default value: None
indicator	str	Explanation: Error indicator returned by the OBS server. Default value: None
hostId	str	Explanation: Requested server ID. If the value of status is less than 300 , this parameter is left blank. Default value: None
resource	str	Explanation: Error source (a bucket or an object). If the value of status is less than 300 , this parameter is left blank. Default value: None
header	list	Explanation: Response header list, composed of tuples. Each tuple consists of two elements, respectively corresponding to the key and value of a response header. Default value: None
body	object	Explanation: Result content returned after the operation is successful. If the value of status is larger than 300 , the value of body is null. The value varies with the API being called. For details, see Bucket-Related APIs (SDK for Python) and Object-Related APIs (SDK for Python) . Default value: None

Code Examples

This example uploads and downloads an encrypted file using SSE-KMS.

```
from obs import ObsClient
from obs import PutObjectHeader, GetObjectHeader
```

```
from obs import SseKmsHeader
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    put_headers = PutObjectHeader()
    # Specify the SSE-KMS encryption header for the object upload request.
    put_headers.sseHeader = SseKmsHeader.getInstance()

    bucketName = "examplebucket"
    # Specify an object name (the name displayed after the file is uploaded to the bucket).
    objectKey = "objectname"
    # Specify the full path of the file or folder to be uploaded, for example, aa/bb.txt or aa/.
    file_path = 'localfile'
    # Upload the object.
    resp = obsClient.putFile(bucketName, objectKey, file_path, headers=put_headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put File Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Put File Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)

# -----

    get_headers = GetObjectHeader()
    # Specify the SSE-KMS decryption header for the object download request.
    get_headers.sseHeader = SseKmsHeader.getInstance()

    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the full path (localfile as an example) to which objects are downloaded. The full path contains
    # the local file name.
    downloadPath = 'localfile'
    # Download the object.
    resp2 = obsClient.getObject(bucketName, objectKey, downloadPath, headers=get_headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp2.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp2.requestId)
    else:
        print('Get Object Failed')
        print('requestId:', resp2.requestId)
        print('errorCode:', resp2.errorCode)
        print('errorMessage:', resp2.errorMessage)
except:
    print(traceback.format_exc())
```

This example uploads and downloads an encrypted file using SSE-C.

```
from obs import ObsClient
from obs import PutObjectHeader, GetObjectHeader
from obs import SseCHeader
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    put_headers = PutObjectHeader()
    # Specify the SSE-C encryption header for the object upload request. encryption indicates the encryption
    # method and key indicates the SSE-C key generated by the AES 256 algorithm.
    put_headers.sseHeader = SseCHeader(encryption='AES256', key='your sse-c key generated by AES-256
    algorithm')

    bucketName = "examplebucket"
    # Specify an object name (the name displayed after the file is uploaded to the bucket).
    objectKey = "objectname"
    # Specify the full path of the file or folder to be uploaded, for example, aa/bb.txt or aa/.
    file_path = 'localfile'
    # Upload the object.
    resp = obsClient.putFile(bucketName, objectKey, file_path, headers=put_headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put File Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Put File Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)

# -----

    get_headers = GetObjectHeader()
    # Specify the SSE-C decryption header for an object download request. The key used here must be the
    # one used for uploading the object.
    get_headers.sseHeader = SseCHeader(encryption='AES256', key='your sse-c key generated by AES-256
    algorithm')

    bucketName = "examplebucket"
    objectKey = "objectname"
    # Specify the full path (localfile as an example) to which objects are downloaded. The full path contains
    # the local file name.
    downloadPath = 'localfile'
    # Download the object.
    resp2 = obsClient.getObject(bucketName, objectKey, downloadPath, headers=get_headers)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp2.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp2.requestId)
    else:
        print('Get Object Failed')
```

```
print('requestId:', resp2.requestId)
print('errorCode:', resp2.errorCode)
print('errorMessage:', resp2.errorMessage)
except:
    print(traceback.format_exc())
```

11.4 Static Website Hosting (SDK for Python)

Function

This API uploads the files of the static website to your bucket in OBS as objects and configures the **public-read** permission on the files, and then configures 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.

Restrictions

- To upload an object, you must have the **obs:object:PutObject** permission.

Uploading a Website File to a Bucket

1. Upload the website files to your bucket in OBS as objects and set the MIME type for the objects.
2. Set the object ACL to public read.
3. Access the objects using a browser.

This example uploads an HTML website file to a bucket and grants the public read permission for the object to implement static website hosting.

```
from obs import ObsClient
import os
from obs import PutObjectHeader
from obs import HeadPermission
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = 'bucketname'
    # Specify a website file name.
    objectKey = 'test.html'
    # Specify the path of a local HTML website file.
    file_path = 'localfile.html'
```

```
headers = PutObjectHeader()
# Specify the MIME type for the object.
headers.contentType = 'text/html'

# Upload the object.
resp = obsClient.putFile(bucketName, objectKey, file_path, headers=headers)

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Put File Succeeded')
    print('requestId:', resp.requestId)
    # Set the object ACL to public read.
    resp2 = obsClient.setObjectAcl(bucketName, objectKey, aclControl=HeadPermission.PUBLIC_READ)
    if resp2.status < 300:
        print('Set Object Acl Succeeded')
        print('requestId:', resp2.requestId)
    else:
        print('Set Object Acl Failed')
        print('requestId:', resp2.requestId)
        print('errorCode:', resp2.errorCode)
        print('errorMessage:', resp2.errorMessage)
else:
    print('Put File Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Put File Failed')
    print(traceback.format_exc())
```

NOTE

You can use **<https://bucketname.your-endpoint/test.html>** in a browser to access files hosted using the sample code.

Configuring Static Website Hosting

This example configures static website hosting for bucket **examplebucket**.

```
from obs import ObsClient
from obs import WebsiteConfiguration
from obs import IndexDocument
from obs import ErrorDocument
from obs import RoutingRule
from obs import Condition
from obs import Redirect
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify an error page when a 4XX error occurs.
    errorDocument = ErrorDocument(key='error.html')
    # Specify a default page.
```

```
indexDocument = IndexDocument(suffix='index.html')
# Specify a rule for redirecting requests to NotFound.html if the status code is 404.
routingRule1 = RoutingRule(condition=Condition(httpErrorCodeReturnedEquals=404),
                           redirect=Redirect(protocol='http', replaceKeyWith='NotFound.html'))
# Configure the redirection rules in list format. Multiple rules can be configured.
routingRules = [routingRule1]
bucketName = "examplebucket"
# Configure static website hosting for the bucket.
resp = obsClient.setBucketWebsite(bucketName,
                                   WebsiteConfiguration(errorDocument=errorDocument,
                                                         indexDocument=indexDocument,
                                                         routingRules=routingRules))

# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Set Bucket Website Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Set Bucket Website Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Website Failed')
    print(traceback.format_exc())
```

This example configures redirection for all requests.

```
from obs import ObsClient
import os
import traceback
from obs import WebsiteConfiguration
from obs import RedirectAllRequestTo

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = 'bucketname'
    # Configure redirection for all requests.
    resp = obsClient.setBucketWebsite(bucketName,
                                       WebsiteConfiguration(
                                           redirectAllRequestTo=RedirectAllRequestTo(hostName='www.example.com',
                                                                                       protocol='http')))
    if resp.status < 300:
        print('Set Bucket Website Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Bucket Website Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Set Bucket Website Failed')
    print(traceback.format_exc())
```

Viewing Static Website Hosting

This example returns the static website hosting configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
# hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
# environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
# security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName="examplebucket"
    # Obtain the static website configuration of the bucket.
    resp = obsClient.getBucketWebsite(bucketName)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Get Bucket Website Succeeded')
        print('requestId:', resp.requestId)
        if resp.body.redirectAllRequestTo:
            print('redirectAllRequestTo.hostName:', resp.body.redirectAllRequestTo.hostName,
                  ',redirectAllRequestTo.protocol:', resp.body.redirectAllRequestTo.protocol)
        if resp.body.indexDocument:
            print('indexDocument.suffix:', resp.body.indexDocument.suffix)
        if resp.body.errorDocument:
            print('errorDocument.key:', resp.body.errorDocument.key)
        if resp.body.routingRules:
            index = 1
            for rout in resp.body.routingRules:
                print('routingRule[', index, ':')
                index += 1
                print('condition.keyPrefixEquals:', rout.condition.keyPrefixEquals,
                      ',condition.httpErrorCodeReturnedEquals:', rout.condition.httpErrorCodeReturnedEquals)
                print('redirect.protocol:', rout.redirect.protocol, 'redirect.hostName:', rout.redirect.hostName,
                      ',redirect.replaceKeyPrefixWith:', rout.redirect.replaceKeyPrefixWith,
                      ',redirect.replaceKeyWith:', rout.redirect.replaceKeyWith, 'redirect.httpRedirectCode:',
                      rout.redirect.httpRedirectCode)
            else:
                print('Get Bucket Website Failed')
                print('requestId:', resp.requestId)
                print('errorCode:', resp.errorCode)
                print('errorMessage:', resp.errorMessage)
    except:
        print('Get Bucket Website Failed')
        print(traceback.format_exc())
```

Deleting Static Website Hosting

This example deletes the static website hosting configuration of bucket **examplebucket**.

```
from obs import ObsClient
import os
import traceback
```



```
# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    # Delete the static website hosting configuration of the bucket.
    resp = obsClient.deleteBucketWebsite(bucketName)

    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Delete Bucket Website Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Delete Bucket Website Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Delete Bucket Website Failed')
    print(traceback.format_exc())
```

11.5 User-defined Headers (SDK for Python)

Function

When calling an API, you can configure user-defined headers to meet specific needs. The SDK will automatically calculate the signature for the specified headers if needed.

Method:

You can add the specified headers in **extensionHeaders** in the dictionary format.

Code Examples

This example configures user-defined headers to use the single-connection bandwidth throttling function for downloading object **objectname** from bucket **examplebucket** at a limited rate.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
```

```
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Configure the download rate limit by specifying x-obs-traffic-limit, in bits. The value range is from
819200 (100 KB) to 838860800 (100 MB). 819200 is used as an example.
    extensionHeaders = {'x-obs-traffic-limit': 819200}
    # Specify the full path (localfile as an example) to which objects are downloaded. The full path contains
    the local file name.
    downloadPath = 'localfile'
    # Download the object at a limited rate.
    resp = obsClient.getObject(bucketName, objectKey, downloadPath, extensionHeaders=extensionHeaders)

    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Get Object Succeeded')
        print('requestId:', resp.requestId)
        print('url:', resp.body.url)
    else:
        print('Get Object Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print('Get Object Failed')
    print(traceback.format_exc())
```

This example configures a user-defined header to implement the upload callback.

```
from obs import ObsClient
import os
import traceback
from urllib.parse import quote
import base64
import json

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Specify a protocol.
    protocol = 'http://'

    # Specify the callback address. If the URL contains any special characters or full-width characters, they
    must be URL-encoded using quote(str).
    callbackUrl1 = protocol + quote("www.example.com/callback1")
    callbackUrl2 = protocol + quote ("www.example.com/full-width-characters?key=object-name-in-full-
width-characters")
    # (Optional) Specify the value of the host header carried in the callback request. If this parameter is not
    specified, the value of host parsed from callbackUrl is used.
    callbackHost = 'www.example.com'
    # Specify the body of the callback request.
    callbackBody = 'key=$(key)&override=$(override)&size=$(size)&bucket=$(bucket)&etag=$(etag)'
```

```
# Configure the upload callback.
callBackPolicy = {"callbackBody": callbackBody, "callbackUrl": callbackUrl1 + ';' + callbackUrl2,
                  "callbackHost": callbackHost}

# Configure the custom headers by specifying extensionHeaders. The input parameters are in the
dictionary format.
# Convert callBackPolicy to a JSON string and then to binary using json.dumps().encode(). Then,
encode the results using Base64 (base64.b64encode()) and convert the encoded data which is in binary
mode to a string using str(b'str', "utf-8").
extensionHeaders = {'x-obs-callback': str(base64.b64encode(json.dumps(callBackPolicy).encode()),
"utf-8"))

bucketName = 'your-bucketName'
objectKey = 'example.txt'
content = 'Hello OBS'
# Upload the text and perform the upload callback.
resp = obsClient.putContent(bucketName, objectKey, content, extensionHeaders=extensionHeaders)

# If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
if resp.status < 300:
    print('Put Content Succeeded')
    print('requestId:', resp.requestId)
    print('etag:', resp.body.etag)
else:
    print('Put Content Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print('Put Content Failed')
    print(traceback.format_exc())
```

This example changes the expiration time of an object using a user-defined header.

```
from obs import ObsClient
import os
import traceback

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using
hard coding may result in leakage.
# Obtain an AK and SK pair on the management console.
ak = os.getenv("AccessKeyId")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from
environment variables.
# security_token = os.getenv("SecurityToken")
# Set server to the endpoint of the region where the bucket is located.
server = "https://your-endpoint"

# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify
security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # x-obs-expires indicates how many days after the last modification the object expires. This example
configures 3 days.
    extensionHeaders = {'x-obs-expire': 3}
    # Configure the object metadata.
    resp = obsClient.setObjectMetadata(bucketName, objectKey, extensionHeaders=extensionHeaders)

    # If status code 2xx is returned, the API was called successfully. Otherwise, the call failed.
    if resp.status < 300:
        print('Set Object Metadata Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Object Metadata Failed')
        print('requestId:', resp.requestId)
```

```
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
    except:
        print('Set Object Metadata Failed')
        print(traceback.format_exc())
```

12 Troubleshooting (SDK for Python)

12.1 HTTP Status Codes (SDK for Python)

The OBS server complies with the HTTP standard. After an API is called, the OBS server returns a standard HTTP status code. The following tables list the categories of HTTP status codes and the common HTTP status codes in OBS.

- Categories of HTTP status codes

Category	Description
1XX	Informational response. A request is received by the server and the server requires the requester to continue the operation. This category is usually invisible to the client.
2XX	Success. The operation is received and processed successfully.
3XX	Redirection. Further operations to complete the request are required.
4XX	Client errors. The request contains a syntax error, or the request cannot be implemented.
5XX	Server errors. An error occurs when the server is processing the request.

- Common HTTP status codes in OBS and their meanings

HTTP Status Code	Description	Possible Cause
400 Bad Request	Incorrect request parameter.	- Invalid request parameter. - The consistency check fails after the client request carries MD5. - An invalid parameter is transferred when the SDK is used. - An invalid bucket name is used.
403 Forbidden	Access denied.	- The signature of the request does not match. Generally, the error is caused by incorrect AK/SK. - The account does not have the permission to access the requested resource. - The account is in arrears. - The bucket space is insufficient when a quota is set for the bucket. - Invalid AK - The time difference between the client and the server is too large. That is, the time of the server where the client is located is not synchronized with the time of the NTP service.
404 Not Found	The requested resource does not exist.	- The bucket does not exist. - The object does not exist. - The bucket policy configuration does not exist. For example, the bucket CORS configuration or bucket policy configuration does not exist. - The multipart upload does not exist.
405 Method Not Allowed	The request method is not supported.	The requested method or feature is not supported in the region where the bucket resides.

HTTP Status Code	Description	Possible Cause
408 Request Timeout	Request timed out.	The socket connection between the server and client timed out.
409 Conflict	Request conflicts occur.	• Buckets of the same name are created in different regions. • Deletion of a non-empty bucket is attempted.
500 Internal Server Error	Internal server error.	Internal server error.
503 Service Unavailable	Service unavailable.	The server cannot be accessed temporarily.

12.2 OBS Server-Side Error Codes (SDK for Python)

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 and its HTTP status code.

HTTP Status Code	Error Code	Error Message	Solution
301 Moved Permanently	PermanentRedirect	The requested bucket can be accessed only through the specified address. Send subsequent requests to the address.	Send the request to the returned redirection address.
301 Moved Permanently	WebsiteRedirect	The website request lacks bucketName .	Put the bucket name in the request and try again.
307 Moved Temporarily	TemporaryRedirect	Temporary redirection. If the DNS is updated, the request is redirected to the bucket.	The system automatically redirects the request or sends the request to the redirection address.

HTTP Status Code	Error Code	Error Message	Solution
400 Bad Request	BadDigest	The specified value of Content-MD5 does not match the value received by OBS.	Check whether the MD5 value carried in the header is the same as that calculated by the message body.
400 Bad Request	BadDomainName	Invalid domain name.	Use a valid domain name.
400 Bad Request	BadRequest	Invalid request parameter.	Modify the parameter according to the error details returned in the message body.
400 Bad Request	CustomDomainAlreadyExist	The configured domain already exists.	It has been configured and does not need to be configured again.
400 Bad Request	CustomDomainNotExist	Delete the domain that does not exist.	It is not configured or has been deleted. You do not need to delete it.
400 Bad Request	EntityTooLarge	The size of the object uploaded using the POST method exceeds the upper limit.	Modify the conditions specified in the policy when posting the object or reduce the object size.
400 Bad Request	EntityTooSmall	The size of the object uploaded using the POST method does not reach the lower limit.	Modify the conditions specified in the policy when posting the object or increase the object size.

HTTP Status Code	Error Code	Error Message	Solution
400 Bad Request	IllegalLocation-ConstraintException	A request without Location is sent for creating a bucket in a non-default region.	Send the bucket creation request to the default region, or send the request with the Location of the non-default region.
400 Bad Request	IncompleteBody	No complete request body is received due to network or other problems.	Upload the object again.
400 Bad Request	IncorrectNumberOfFilesInPostRequest	Each POST request must contain one file to be uploaded.	Carry a file to be uploaded.
400 Bad Request	InvalidArgument	Invalid parameter.	Modify the parameter according to the error details returned in the message body.
400 Bad Request	InvalidBucket	The bucket to be accessed does not exist.	Try another bucket name.
400 Bad Request	InvalidBucketName	The bucket name specified in the request is invalid, which may have exceeded the maximum length, or contain special characters that are not allowed.	Try another bucket name.
400 Bad Request	InvalidLocation-Constraint	The specified Location in the bucket creation request is invalid or does not exist.	Correct the Location in the bucket creation request.

HTTP Status Code	Error Code	Error Message	Solution
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.	Specify the correct parts and entity tags.
400 Bad Request	InvalidPartOrder	Parts are not listed in ascending order by part number.	Sort the parts in ascending order and merge them again.
400 Bad Request	InvalidPolicyDocument	The content of the form does not meet the conditions specified in the policy document.	Modify the policy in the constructed form according to the error details in the message body and try again.
400 Bad Request	InvalidRedirectLocation	Invalid redirect location.	Specify the correct IP address.
400 Bad Request	InvalidRequest	Invalid request.	Modify the parameter according to the error details returned in the message body.
400 Bad Request	InvalidRequestBody	The request body is invalid. The request requires a message body but no message body is uploaded.	Upload the message body in the correct format.
400 Bad Request	InvalidTargetBucketForLogging	The delivery group has no ACL permission for the target bucket.	Configure the target bucket ACL and try again.
400 Bad Request	KeyTooLongError	The provided key is too long.	Use a shorter key.

HTTP Status Code	Error Code	Error Message	Solution
400 Bad Request	MalformedACLError	The provided XML file has syntax errors or does not meet the format requirements.	Use the correct XML format to retry.
400 Bad Request	MalformedError	The XML format in the request is incorrect.	Use the correct XML format to retry.
400 Bad Request	MalformedLoggingStatus	The XML format of Logging is incorrect.	Use the correct XML format to retry.
400 Bad Request	MalformedPolicy	The bucket policy does not pass.	Modify the bucket policy according to the error details returned in the message body.
400 Bad Request	MalformedQuotaError	The Quota XML format is incorrect.	Use the correct XML format to retry.
400 Bad Request	MalformedXML	An XML file of a configuration item is in incorrect format.	Use the correct XML format to retry.
400 Bad Request	MaxMessageLengthExceeded	Copying an object does not require a message body in the request.	Remove the message body and retry.
400 Bad Request	MetadataTooLarge	The size of the metadata header has exceeded the upper limit.	Reduce the size of the metadata header.
400 Bad Request	MissingRegion	The region information is missing in the request, and the default region is required in the system.	Carry the region information in the request.
400 Bad Request	MissingRequestBodyError	This error code is returned after you send an empty XML file.	Provide the correct XML file.

HTTP Status Code	Error Code	Error Message	Solution
400 Bad Request	MissingRequired-Header	A required header is missing in the request.	Provide the required header.
400 Bad Request	MissingSecurity-Header	A required header is missing in the request.	Provide the required header.
400 Bad Request	TooManyBuckets	You have attempted to create more buckets than allowed.	Delete some buckets and try again.
400 Bad Request	TooManyCustomDomains	Too many user accounts are configured.	Delete some user accounts and try again.
400 Bad Request	TooManyWrongSignatures	The request is rejected due to high-frequency errors.	Replace AK and try again.
400 Bad Request	UnexpectedContent	The request requires a message body which is not carried by the client, or the request does not require a message body but the client carries the message body.	Try again according to the instruction.
400 Bad Request	UserKeyMustBeSpecified	This operation is only available to special users.	Contact the technical support.
403 Forbidden	AccessDenied	Access denied, because the request does not carry a date header or the header format is incorrect.	Provide a correct date header in the request.

HTTP Status Code	Error Code	Error Message	Solution
403 Forbidden	AccessForbidden	Insufficient permissions. No CORS rule is configured for the bucket, or the CORS rule is not matched.	Modify the CORS configuration of the bucket or send the matched OPTIONS request based on the CORS configuration of the bucket.
403 Forbidden	AllAccessDisabled	The user has no permission to perform a specific operation. The bucket name is forbidden.	Try another bucket name.
403 Forbidden	DeregisterUserId	The user has been deregistered.	Top up or re-register.
403 Forbidden	InArrearOrInsufficientBalance	The user account is in arrears, or the account balance is insufficient.	Top up the account.
403 Forbidden	InsufficientStorageSpace	Insufficient storage space.	If the quota is exceeded, increase quota or delete some objects.
403 Forbidden	InvalidAccessKeyId	The access key ID provided by the customer does not exist in the system.	Provide correct access key ID.
403 Forbidden	RequestTimeTooSkewed	The request time and the server's time differ a lot.	Check whether the difference between the client time and the current time is too large.
403 Forbidden	SignatureDoesNotMatch	The provided signature in the request does not match the signature calculated by OBS.	Check your secret access key and signature calculation method.

HTTP Status Code	Error Code	Error Message	Solution
403 Forbidden	Unauthorized	You have not been authenticated in real name.	Authenticate your real name and try again.
404 Not Found	NoSuchBucket	The bucket does not exist.	Create a bucket and perform the operation again.
404 Not Found	NoSuchBucketPolicy	No bucket policy exists.	Configure a bucket policy.
404 Not Found	NoSuchCORSConfiguration	No CORS configuration exists.	Configure CORS first.
404 Not Found	NoSuchCustomDomain	The requested user domain does not exist.	Set a user domain first.
404 Not Found	NoSuchKey	The specified key does not exist.	Upload the object first.
404 Not Found	NoSuchLifecycleConfiguration	The requested Lifecycle does not exist.	Configure a lifecycle rule first.
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.	Use the existing part or reinitialize the part.
404 Not Found	NoSuchVersion	The specified version ID does not match any existing version.	Use a correct version ID.
404 Not Found	NoSuchWebsiteConfiguration	The requested website does not exist.	Configure the website first.

HTTP Status Code	Error Code	Error Message	Solution
405 Method Not Allowed	MethodNotAllowed	The specified method is not allowed against the requested resource. The message "Specified method is not supported." is returned.	The method is not allowed.
408 Request Timeout	RequestTimeout	The socket connection to the server has no read or write operations within the timeout period.	Check the network and try again, or contact technical support.
409 Conflict	BucketAlreadyExists	The requested bucket name already exists. The bucket namespace is shared by all users of OBS. Try again with another bucket name.	Try another bucket name.
409 Conflict	BucketAlreadyOwnedByYou	Your previous request for creating the named bucket succeeded and you already own it.	You do not need to create the bucket again.
409 Conflict	BucketNotEmpty	The bucket that you tried to delete is not empty.	Delete the objects in the bucket and then delete the bucket.
409 Conflict	OperationAborted	A conflicting operation is being performed on this resource. Try again later.	Try again later.

HTTP Status Code	Error Code	Error Message	Solution
409 Conflict	ServiceNotSupported	The request method is not supported by the server.	Not supported by the server. Contact technical support.
411 Length Required	MissingContentLength	The HTTP header Content-Length is not provided.	Provide the Content-Length header.
412 Precondition Failed	PreconditionFailed	At least one of the specified preconditions is not met.	Modify according to the condition prompt in the returned message body.
416 Client Requested Range Not Satisfiable	InvalidRange	The requested range cannot be obtained.	Replace the range value and try again.
500 Internal Server Error	InternalServerError	An internal error occurs. Try again later.	Contact the technical support.
501 Not Implemented	ServiceNotImplemented	The request method is not implemented by the server.	Not supported currently. Contact the technical support.
503 Service Unavailable	ServiceUnavailable	The server is overloaded or has internal errors.	Try again later or contact the technical support.
503 Service Unavailable	SlowDown	Reduce your request frequency.	Reduce your request frequency.

12.3 Log Analysis (SDK for Python)

Log Configuration

OBS Python SDK provides the logging function based on the Python log library. You can call **ObsClient.initLog** to enable and configure logging. Sample code is as follows:

```
# Import the module.
from obs import LogConf
from obs import ObsClient

# Create an instance of ObsClient.
obsClient = ObsClient(
    access_key_id=os.getenv("AccessKeyID"),
    secret_access_key=os.getenv("SecretAccessKey"),
    server='https://your-endpoint'
```



```
)  
  
# Specify the path to the log configuration file and initialize logs of ObsClient.  
obsClient.initLog(LogConf('./log.conf'), 'obs_logger');
```

NOTE

- The logging function is disabled by default. You need to enable it manually.
- The log configuration file example (**log.conf**) is included in the OBS Python SDK development package. Modify parameters in **log.conf** as needed.

NOTICE

The log module of the OBS Python SDK is thread secure but not process secure. If ObsClient is used in multi-process scenarios, you must configure an independent log path for each instance of ObsClient to prevent conflicts when multiple processes write logs concurrently.

Log Format

The SDK log format is: *Log time|Process ID|Thread number|Log level|Log content*. The following are example logs:

```
2017-11-06 13:46:54,936|process:6100|thread:12700|DEBUG|HTTP(s)+XML|OBS_LOGGER|__parse_xml,188|  
http response result:status:200,reason:OK,code:None,message:None,headers:[('id-2',  
'LgOKocHfuHe0rFSUHS6LcChzcoYes0luPgqxhUfCP58xp3MZh2n4YKRPPABV8GEK'), ('connection', 'close'),  
('request-id', '0001AFF8E60000015F8FDA1EA5AE04E3'), ('date', 'Mon, 06 Nov 2017 05:42:37 GMT'),  
('content-type', 'application/xml')]]  
2017-11-06 13:46:54,937|process:6100|thread:12700|INFO|HTTP(s)+XML|OBS_LOGGER|doClose,349|server  
inform to close connection|  
2017-11-06 13:46:54,937|process:6100|thread:12700|INFO|HTTP(s)+XML|OBS_LOGGER|wrapper,59|  
listBuckets cost 56 ms|
```

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.

Log level description:

- **DEBUG**: Debugging level. If this level is set, all logs will be printed.
- **INFO**: Information level. If this level is set, logs at the **WARNING** level and the time consumed for each HTTP/HTTPS request will be printed.
- **WARNING**: Warning level. If this level is set, logs at the **ERROR** level and some critical events will be printed.
- **ERROR**: Error level. If this level is set, only error information will be printed.

NOTE

In the configuration file, **LogFileLevel** is used to specify the log level for log files, and **PrintLogLevel** is used to specify the log level for the console.

13 FAQs (SDK for Python)

13.1 How Do I Make an Object Accessible to Anonymous Users? (SDK for Python)

To do this, perform the following steps:

1. Set the object ACL to public read by referring to [Configuring an Object ACL \(SDK for Python\)](#).
2. Obtain the URL of the object by referring to [How Do I Obtain an Object URL? \(Python SDK\)](#) and provide it to anonymous users.
3. An anonymous user can access the object by entering the URL on a browser.

13.2 How Do I Obtain an Object URL? (Python SDK)

[After setting an ACL to grant anonymous users the read permission for an object in a bucket](#), you can download this object using its URL. Methods to obtain the object URL are as follows:

Method 1: Query by calling the API. After an object is uploaded by calling **ObsClient.putContent** or **ObsClient.putFile**, **PutContentResponse** is returned. You can call **objectUrl** to obtain the URL of the uploaded object. Sample code is as follows:

```
# Import the module.
from obs import ObsClient

# Create an instance of ObsClient.
obsClient = ObsClient(
    access_key_id=os.getenv("AccessKeyId"),
    secret_access_key=os.getenv("SecretAccessKey"),
    server='https://your-endpoint'
)
resp = obsClient.putContent('bucketname', 'objectname', content='Hello OBS')

if resp.status < 300:
    print('requestId:', resp.requestId)
    print('objectUrl:', resp.body.objectUrl)
else:
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
```

Method 2: Compose the URL in the format of **https://Bucket name.Domain name/Directory level/Object name**.

 NOTE

- If the object resides in the root directory of a bucket, its URL does not contain a directory level.

13.3 How Do I Improve the Uploading Speed of Large Files over the Public Network? (Python SDK)

If a file exceeds 100 MB, you are advised to upload the file using multipart upload.

Multipart upload refers to splitting an object into multiple parts and uploading them separately. Each part is a contiguous portion of the object's data. You can upload parts in any sequence. A part can be reloaded after an upload failure, without affecting other parts. Uploading multiple parts of an object using multiple threads concurrently can greatly improve the transmission efficiency.

For details about the sample code, see [APIs Related to Multipart Upload \(SDK for Python\)](#).

13.4 Setting an Object Expiration Time (SDK for Python)

This example sets the object expiration time using a header when uploading a file stream.

```
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Read a file stream.
    content = open('localfile', 'rb')
    bucketName = "examplebucket"
    objectKey = "objectname"
    header=PutObjectHeader()
    #Set the expiration time.
    header.expires=10
    # Upload the file stream.
    resp = obsClient.putContent(bucketName, objectkey, content,headers=header)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Put Content Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Put Content Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print(traceback.format_exc())
```

This example sets the object expiration time using a user-defined header when uploading a file stream.

```
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Read a file stream.
    content = open('localfile', 'rb')
    bucketName = "examplebucket"
```

```
objectKey = "objectname"
header=PutObjectHeader()
# Use a user-defined header to set the expiration time.
extensionHeaders = {'x-obs-expires': 30}
# Upload the file stream.
resp = obsClient.putContent(bucketName, objectkey, content,extensionHeaders=extensionHeaders)
# If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
if resp.status < 300:
    print('Put Content Succeeded')
    print('requestId:', resp.requestId)
else:
    print('Put Content Failed')
    print('requestId:', resp.requestId)
    print('errorCode:', resp.errorCode)
    print('errorMessage:', resp.errorMessage)
except:
    print(traceback.format_exc())
```

This example sets the expiration time for an uploaded object.

```
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    bucketName = "examplebucket"
    objectKey = "objectname"
    # Use a user-defined header to set the expiration time.
    extensionHeaders = {'x-obs-expires': 30}
    # Configure metadata for the object.
    resp = obsClient.setObjectMetadata(bucketName, objectkey, extensionHeaders=extensionHeaders)
    # If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
    if resp.status < 300:
        print('Set Object Metadata Succeeded')
        print('requestId:', resp.requestId)
    else:
        print('Set Object Metadata Failed')
        print('requestId:', resp.requestId)
        print('errorCode:', resp.errorCode)
        print('errorMessage:', resp.errorMessage)
except:
    print(traceback.format_exc())
```

13.5 What Is Content-Type (MIME)? (Python SDK)

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.

File Name Extension	Content-Type	File Name Extension	Content-Type
.* (binary stream, unknown file type)	application/octet-stream	.7z	application/x-7z-compressed
.001	application/x-001	.301	application/x-301
.323	text/h323	.906	application/x-906
.907	drawing/907	.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
.asa	text/asa	.asf	video/x-ms-asf
.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
.cut	application/x-cut	.dbf	application/x-dbf
.dbm	application/x-dbm	.dbx	application/x-dbx

File Name Extension	Content-Type	File Name Extension	Content-Type
.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	.dot	application/msword
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.ico	image/x-icon	.ico	application/x-ico
.iff	application/x-iff	.ig4	application/x-g4
.igs	application/x-igs	.iii	application/x-iphone
.img	application/x-img	.ins	application/x-internet-signup
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.mocha	application/x-javascript	.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	.mxx	application/x-mxx
.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	.pdf	application/pdf
.pdx	application/vnd.adobe.pdx	.pfx	application/x-pkcs12
.pgl	application/x-pgl	.pic	application/x-pic

File Name Extension	Content-Type	File Name Extension	Content-Type
.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
.ppa	application/vnd.ms-powerpoint	.ppm	application/x-ppm
.pps	application/vnd.ms-powerpoint	.ppt	application/vnd.ms-powerpoint
.ppt	application/x-ppt	.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
.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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- realtext
.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	.sit	application/x- stuffit
.slb	application/x-slb	.sld	application/x-sld
.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	text/xml

File Name Extension	Content-Type	File Name Extension	Content-Type
.swf	application/x-shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tiff	image/tiff
.ttf	font/otf	.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-vpeg005	.vsd	application/vnd.visio
.vsd	application/x-vsdx	.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

File Name Extension	Content-Type	File Name Extension	Content-Type
.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
.xdp	application/vnd.adobe.xdp	.xdr	text/xml
.xfd	application/vnd.adobe.xfd	.xdf	application/vnd.adobe.xdf
.xhtml	text/html	.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
.xls	application/x-xls	.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_b	application/x-x_b
.sis	application/vnd.symbian.install	.sisx	application/vnd.symbian.install
.x_t	application/x-x_t	.ipa	application/vnd.iphone
.apk	application/vnd.android.package-archive	.xap	application/x-silverlight-app
.zip	application/zip	.rar	application/x-rar-compressed

13.6 How Do I Get My Account ID and IAM User ID? (SDK for Python)

Obtaining the Account ID and User ID

When calling APIs, you may need to specify the account ID (**DomainID**) and user ID (**UserID**) in some requests. You need to obtain them from the console in advance. To obtain them from the console, do as follows:

Step 1 Log in to the management console.

Step 2 Hover the mouse pointer over the username and choose **My Credentials** from the drop-down list.

On the **My Credentials** page, view the account ID and user ID.

----End