API Reference

OBS Node.js SDK

Issue	01
Date	2023-03-14

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to “Vul. Response Process”. For details about the policy, see the following website:https://www.huawei.com/en/psirt/vul-response-process
For enterprise customers who need to obtain vulnerability information, visit:https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Overview

This document describes all APIs of OBS (Object Storage Service) Node.js SDK, including the API description, method definition, and parameter description.

For details about the end-to-end use (such as installation, initialization, development, and FAQs) of OBS Node.js SDK, application scenarios of APIs, and code examples in different scenarios, see the **Object Storage Service Node.js SDK Developer Guide**.

2 Initialization

2.1 Initializing an ObsClient Instance

API Description

ObsClient functions as the Node.js client for accessing OBS. It offers callers a series of APIs for interaction with OBS and is used for managing and operating resources, such as buckets and objects, stored in OBS.

Method Definition

```
1. Constructor form: ObsClient(parameter)
2. Factory method form: ObsClient.factory(parameter)
```

Parameter Description

Field	Type	Optional or Mandatory	Description
access_key_id	String	Optional	AK
secret_access_key	String	Optional	SK
server	String	Mandatory	Endpoint for accessing OBS, which contains the protocol type, domain name (or IP address), and port number. For example, https://your-endpoint:443.
max_retry_count	Number	Optional	Maximum number of retries when an HTTP/HTTPS connection is abnormal. The default value is **3**.

Field	Type	Optional or Mandatory	Description
timeout	Number	Optional	Timeout period (in seconds) of an HTTP/HTTPS request. The default value is **60**.
ssl_verify	Boolean or String	Optional	Whether to verify the server certificate. Possible values are: • Path to the server-side root certificate file in **.pem** format • **true**: The default CAs are used to verify the server-side certificate. • **false**: The server-side certificates will not be verified. The default value is **false**.
long_conn_param	Number	Optional	Persistent connection mode (in seconds) If the value is equal to or larger than **0**, the persistent connection mode is enabled and this value is used as the **initial delay** of the TCP Keep-Alive packets. By default, this parameter is left blank, which indicates that persistent connection mode is disabled.
is_cname	Boolean	Optional	Whether to use self-defined domain name to access OBS. The default value is **false**.

Sample Code

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

// Create an ObsClient instance.
var obsClient = new ObsClient({
      //Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
      //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
      access_key_id: process.env.ACCESS_KEY_ID,
      secret_access_key: process.env.SECRET_ACCESS_KEY,
   //EU-Dublin region is used here as an example. Replace it with the one in your actual situation.
      server: 'https://obs.eu-west-101.myhuaweicloud.com'
```



```
    max_retry_count : 1,
    timeout : 20,
    ssl_verify : false,
    long_conn_param : 0,

});
```

2.2 Initializing Logs

API Description

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.

Method Definition

```
ObsClient.initLog(parameter)
```

Parameter Description

Field	Type	Optional or Mandatory	Description
name	String	Optional	Log name. When there are multiple instances of **ObsClient**, this parameter is used to identify log files.
file_full_path	String	Mandatory	Full path to the log file
max_log_size	String	Mandatory	Log file size in bytes
backups	String	Mandatory	Maximum number of retained log files
level	String	Mandatory	Log level. Possible values are **debug**, **info**, **warn**, and **error**.
log_to_console	Boolean	Optional	Whether to print logs to the console

Sample Code

```
obsClient.initLog({
    name: 'client1',
    file_full_path:'./logs/OBS-SDK.log',
    max_log_size:20480,
    backups:10,
    level:'info',
    log_to_console:false
});
```


2.3 Request Objects

Description

Each time you call an API in an instance of **ObsClient**, you need to pass a request object (Object type) as the input. For a bucket-related API, the request object always contains the **Bucket** field to specify the bucket name. For an object-related API, the request object always contains the **Bucket** and **Key** fields to specify the bucket name and object name, respectively.

Parameter Description

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory for object-related APIs	Object name
Others	For details, see chapters "Bucket-Related APIs" and "Object-Related APIs".		

2.4 SDK Common Result Objects

Description

After you call an API in an instance of the **ObsClient** class, a common result object will be returned if no exception is thrown.

Parameter Description

Field		Type	Description
CommonMsg		Object	Common information generated after the API calling is complete, including the HTTP status code and error code
	Status	Number	HTTP status code. If the value is smaller than **300**, the operation succeeds. Otherwise, the operation fails.
	Code	String	Error code returned by the OBS server. If **Status** is smaller than **300**, the value is **null**.
	Message	String	Error description returned by the OBS server. If **Status** is smaller than **300**, the value is **null**.

Field		Type	Description
	HostId	String	Server ID. If **Status** is smaller than **300**, the value is **null**.
	RequestId	String	Request ID returned by the OBS server
	Id2	String	Request ID2 returned by the OBS server
	Indicator	String	Detailed error code returned by the OBS server. If **Status** is smaller than **300**, the value is **null**.
InterfaceResult		Object	Result returned after an operation is successfully performed. If **Status** is smaller than **300**, the value is **null**.
	RequestId	String	Request ID returned by the OBS server
	Id2	String	Request ID2 returned by the OBS server
	Others	For details, see chapters "Bucket-Related APIs" and "Object-Related APIs".	

2.5 Result Returned via a Callback Function

Description

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

Sample Code

```
obsClient.putObject({
    Bucket : 'bucketname',
    Key : 'objectkey',
    Body : 'Hello OBS'
}, (err, result) => {
    if(err){
        console.log('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            if(result.InterfaceResult){
                console.log('Operation Succeed');
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
            console.log('HostId-->' + result.CommonMsg.HostId);
            console.log('RequestId-->' + result.CommonMsg.RequestId);
```



```
        }
    }
});
```

2.6 Result Returned via the Promise Object

Description

ObsClient supports results returned via the **Promise** object. If no exception is caught by the **catch** method of the **Promise** object, the API calling is complete. In such conditions, you need to obtain the HTTP status code from the **SDK common result object** to check whether the operation is successful.

Sample Code

```
obsClient.putObject({
    Bucket : 'bucketname',
    Key : 'objectkey',
    Body : 'Hello OBS'
}).then((result) => {
    if(result.CommonMsg.Status < 300){
        if(result.InterfaceResult){
            console.log('Operation Succeed');
        }
    }else{
        console.log('Code-->' + result.CommonMsg.Code);
        console.log('Message-->' + result.CommonMsg.Message);
        console.log('HostId-->' + result.CommonMsg.HostId);
        console.log('RequestId-->' + result.CommonMsg.RequestId);
    }
}).catch((err) => {
    console.error('Error-->' + err);
});
```


3 Pre-defined Constants

3.1 Permission Types

Module	Access Method	Type	Description
obs.js	ObsClient.enums.PermissionRead	String	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, and multiple object versions in and the bucket, as well as metadata of the bucket. A grantee with this permission for an object can obtain the object content and metadata.
obs.js	ObsClient.enums.PermissionWrite	String	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
obs.js	ObsClient.enums.PermissionReadAcp	String	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission permanently.

Module	Access Method	Type	Description
obs.js	ObsClient.enums.PermissionWriteAcp	String	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission permanently. A grantee with this permission can modify the access control policy and thus the grantee obtains full access permissions.
obs.js	ObsClient.enums.PermissionFullControl	String	A grantee with this permission for a bucket has **PermissionRead**, **PermissionWrite**, **PermissionReadAcp**, and **PermissionWriteAcp** permissions for the bucket. A grantee with this permission for an object has **PermissionRead**, **PermissionWriteAcp**, and **PermissionWriteAcp** permissions for the object.

3.2 Available Grantee Types

Module	Access Method	Type	Description
obs.js	ObsClient.enums.GranteeGroup	String	Grants permissions to user groups.
obs.js	ObsClient.enums.GranteeUser	String	Grants permissions to a single user.

3.3 Available Grantee Groups

Module	Access Method	Type	Description
obs.js	ObsClient.enums.GroupAllUsers	String	All users

Module	Access Method	Type	Description
obs.js	ObsClient.enums.GroupAuthenticatedUsers	String	Authorized users. This constant is deprecated.
obs.js	ObsClient.enums.GroupLogDelivery	String	Log delivery group. This constant is deprecated.

3.4 Pre-defined Access Control Policies

Module	Access Method	Type	Description
obs.js	ObsClient.enums.AclPrivate	String	Private read/write
obs.js	ObsClient.enums.AclPublicRead	String	Public read
obs.js	ObsClient.enums.AclPublicReadWrite	String	Public read/write
obs.js	ObsClient.enums.AclPublicReadDelivered	String	Public read on a bucket as well as objects in the bucket
obs.js	ObsClient.enums.AclPublicReadWriteDelivered	String	Public read/write on a bucket as well as objects in the bucket

3.5 Storage Classes

Module	Access Method	Type	Description
obs.js	ObsClient.enums.StorageClassStandard	String	OBS Standard
obs.js	ObsClient.enums.StorageClassWarm	String	OBS Infrequent Access
obs.js	ObsClient.enums.StorageClassCold	String	OBS Archive

3.6 Restore Options

Module	Access Method	Type	Description
obs.js	ObsClient.enums.RestoreTierExpedited	String	Expedited restoration, which restores an object in 1 to 5 minutes.
obs.js	ObsClient.enums.RestoreTierStandard	String	Standard restoration, which restores an object in 3 to 5 hours.

3.7 Metadata Replication Policy

Module	Access Method	Type	Description
obs.js	ObsClient.enums.CopyMetadata	String	Copies metadata.
obs.js	ObsClient.enums.ReplaceMetadata	String	Replaces metadata.

4 Bucket-Related APIs

4.1 PUT Bucket

API Description

You can use this API to create a bucket and name it as you specify. The created bucket name must be unique in OBS. If a user repeatedly creates buckets with the same name in one region, status code **200** is returned. In other cases, status code **409** is returned. Each user can create a maximum of 100 buckets.

NOTE

Bucket-Related API functions of **ObsClient** are case insensitive. For example, **ObsClient.createBucket** and **ObsClient.CreateBucket** indicate the same function.

Method Definition

```
ObsClient.createBucket
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name. A bucket name must comply with the following rules: • Contains 3 to 63 characters, starts with a digit or letter, and supports only lowercase letters, digits, hyphens (-), and periods (.) • Cannot be an IP-like 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**.
ACL	String	Optional	**Pre-defined access control policy** that can be specified during the bucket creation
StorageClass	String	Optional	Bucket **storage class** that can be specified during the bucket creation
Location	String	Mandatory unless the region where the OBS service resides is not the default region.	Region where a bucket will be created.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.createBucket({
      Bucket : 'bucketname',
      ACL : obsClient.enums.AclPrivate,
      StorageClass : obsClient.enums.StorageClassStandard
}, (err, result) => {
      if(err){
            console.error('Error-->' + err);
      }else{
            if(result.CommonMsg.Status < 300){
                  console.log('RequestId-->' + result.InterfaceResult.RequestId);
            }else{
                  console.log('Code-->' + result.CommonMsg.Code);
                  console.log('Message-->' + result.CommonMsg.Message);
            }
      }
});
```

4.2 GET Buckets

API Description

You can use this API to obtain the bucket list. In the list, bucket names are displayed in lexicographical order.

Method Definition

```
ObsClient.listBuckets
```

Request Parameter

Field	Type	Optional or Mandatory	Description
QueryLocation	Boolean	Optional	Whether to query the bucket location

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
Buckets		Array	Bucket list
	BucketName	String	Bucket name
	CreationDate	String	Creation time of the bucket
	Location	String	Bucket location
Owner		Object	Bucket owner

Field		Type	Description
	ID	String	ID of the domain to which the bucket owner belongs

Sample Code

```
obsClient.listBuckets({
    QueryLocation: true
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Owner:');
            console.log('ID-->' + result.InterfaceResult.Owner.ID);
            console.log('Buckets:');
            for(let i=0;i<result.InterfaceResult.Buckets.Bucket.length;i++){
                console.log('Bucket[' + i + ']:');
                console.log('BucketName-->' + result.InterfaceResult.Buckets[i].BucketName);
                console.log('CreationDate-->' + result.InterfaceResult.Buckets[i].CreationDate);
                console.log('Location-->' + result.InterfaceResult.Buckets[i].Location);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.3 HEAD Bucket

API Description

You can use this API to check whether a bucket exists. If the returned HTTP response code is **200**, the bucket exists. If the returned HTTP response code is **404**, the bucket does not exist.

Method Definition

```
ObsClient.headBucket
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.headBucket({
    Bucket : 'bucketname'
}, (err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('Bucket exists');
        }else if(result.CommonMsg.Status === 404){
            console.log('Bucket does not exist');
        }
    }
});
```

4.4 DELETE Bucket

API Description

You can use this API to delete a bucket. The bucket to be deleted must be empty (containing no objects, noncurrent object versions, or part fragments).

Method Definition

```
ObsClient.deleteBucket
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucket({
      Bucket : 'bucketname'
}, (err, result) => {
      if(err){
            console.error('Error-->' + err);
      }else{
            if(result.CommonMsg.Status < 300){
                  console.log('RequestId-->' + result.InterfaceResult.RequestId);
            }else{
                  console.log('Code-->' + result.CommonMsg.Code);
                  console.log('Message-->' + result.CommonMsg.Message);
            }
      }
});
```

4.5 GET Objects

API Description

You can use this API to list objects in a bucket. By default, a maximum of 1000 objects are listed.

Method Definition

```
ObsClient.listObjects
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Prefix	String	Optional	Prefix that the object names to be listed must contain
Marker	String	Optional	Object name to start with when listing objects in a bucket. All objects following this parameter are listed in the lexicographical order.
MaxKeys	Number	Optional	Maximum number of objects returned. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.

Field	Type	Optional or Mandatory	Description
Delimiter	String	Optional	Character used to group object names. If the object name contains the **Delimiter** parameter, the character string from the first character to the first **Delimiter** parameter in the object name is grouped under a single result element, **CommonPrefix**. (If a prefix is specified in the request, the prefix must be removed from the object name.)

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Location	String	Bucket location
Bucket	String	Bucket name
Delimiter	String	Character used to group object names, which is consistent with that set in the request
IsTruncated	String	Whether all objects are returned for a request. If the field value is **true**, not all objects are returned. If the field value is **false**, all objects are returned.
Prefix	String	Object name prefix, which is consistent with that set in the request
Marker	String	Start position for listing objects, which is consistent with that set in the request
NextMarker	String	Object name to start with upon the next request for listing objects in a bucket
MaxKeys	String	Maximum number of listed objects, which is consistent with that set in the request
Contents	Array	Object list.

Field			Type	Description
	ETag		String	MD5 value of the object (If the object is encrypted using server-side encryption, the ETag is not the MD5 value of the object.)
	Size		String	Object size in bytes
	Key		String	Object name
	LastModified		String	Time when the last modification was made to the object
	Owner		Object	Object owner
		ID	String	ID of the domain to which the object owner belongs
	StorageClass		String	Storage class of the object
	Type		String	Whether the object is an appendable object
CommonPrefixes			Array	List of object name prefixes grouped according to the **Delimiter** parameter (if specified)
	Prefix		String	Object name prefix grouped according to the **Delimiter** parameter

Sample Code

```
obsClient.listObjects({
    Bucket : 'bucketname',
    Prefix : 'prefix',
    MaxKeys : 100
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            for(let j=0;j<result.InterfaceResult.Contents.length;j++){
                console.log('Contents[' + j +  ']:');
                console.log('Key-->' + result.InterfaceResult.Contents[j]['Key']);
                console.log('LastModified-->' + result.InterfaceResult.Contents[j]['LastModified']);
                console.log('ETag-->' + result.InterfaceResult.Contents[j]['ETag']);
                console.log('Size-->' + result.InterfaceResult.Contents[j]['Size']);
                console.log('Owner[ID]-->' + result.InterfaceResult.Contents[j]['Owner']['ID']);
                console.log('StorageClass-->' + result.InterfaceResult.Contents[j]['StorageClass']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.6 GET Object versions

API Description

You can use this API to list versioning objects in a bucket. By default, a maximum of 1000 versioning objects are listed.

Method Definition

```
ObsClient.listVersions
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Prefix	String	Optional	Prefix that the object names to be listed must contain
KeyMarker	String	Optional	Object name to start with when listing versioning objects in a bucket. All objects following this parameter are listed in the lexicographical order of object names.
MaxKeys	Number	Optional	Maximum number of versioning objects returned. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.
Delimiter	String	Optional	Character used to group object names. If the object name contains the **Delimiter** parameter, the character string from the first character to the first **Delimiter** parameter in the object name is grouped under a single result element, **CommonPrefix**. (If a prefix is specified in the request, the prefix must be removed from the object name.)

Field	Type	Optional or Mandatory	Description
VersionIdMarker	String	Optional	Object name to start with when listing versioning objects in a bucket. All versioning objects are listed in the lexicographical order by object name and version ID. This parameter must be used together with **KeyMarker**. If the value of **VersionIdMarker** is not a version ID specified by **KeyMarker**, **VersionIdMarker** does not take effect.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Location	String	Bucket location
Bucket	String	Bucket name
Delimiter	String	Character used to group object names, which is consistent with that set in the request
Prefix	String	Object name prefix, which is consistent with that set in the request
IsTruncated	String	Whether all versioning objects are returned for a request. If the field value is **true**, not all versioning objects are returned. If the field value is **false**, all versioning objects are returned.
KeyMarker	String	Start position for listing versioning objects, which is consistent with that set in the request
VersionIdMarker	String	Version ID to start with for listing versioning objects, which is consistent with that set in the request
NextKeyMarker	String	Object name to start with upon the next request for listing versioning objects in a bucket

Field			Type	Description
NextVersionIdMarker			String	Version to start with upon the next request for listing versioning objects in a bucket. It must be used with the **NextKeyMarker** parameter.
MaxKeys			String	Maximum number of listed versioning objects, which is consistent with that set in the request
Versions			Array	List of versioning objects.
	ETag		String	MD5 value of the object
	Size		String	Object size in bytes
	Key		String	Object name
	VersionId		String	Object version ID
	IsLatest		String	Whether the object is of the latest version. If the parameter value is **true**, the object is of the latest version.
	LastModified		String	Time when the last modification was made to the object
	Owner		Object	Object owner
		ID	String	ID of the domain to which the object owner belongs
	StorageClass		String	Storage class of the object
	Type		String	Whether the object is an appendable object
DeleteMarkers			Array	List of delete markers
	Owner		Object	Object owner
		ID	String	ID of the domain to which the object owner belongs
	Key		String	Object name
	VersionId		String	Object version ID
	IsLatest		String	Whether the object is of the latest version. If the parameter value is **true**, the object is of the latest version.
	LastModified		String	Time when the last modification was made to the object
CommonPrefixes			Array	List of object name prefixes grouped according to the **Delimiter** parameter (if specified)

Field		Type	Description
	Prefix	String	Object name prefix grouped according to the **Delimiter** parameter

Sample Code

```
obsClient.listVersions({
    Bucket : 'bucketname',
    Prefix : 'prefix',
    MaxKeys : 100
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Versions:');
            for(let j=0;j<result.InterfaceResult.Versions.length;j++){
                console.log('Version[' + j +  ']:');
                console.log('Key-->' + result.InterfaceResult.Versions[j]['Key']);
                console.log('VersionId-->' + result.InterfaceResult.Versions[j]['VersionId']);
                console.log('IsLatest-->' + result.InterfaceResult.Versions[j]['IsLatest']);
                console.log('LastModified-->' + result.InterfaceResult.Versions[j]['LastModified']);
                console.log('ETag-->' + result.InterfaceResult.Versions[j]['ETag']);
                console.log('Size-->' + result.InterfaceResult.Versions[j]['Size']);
                console.log('Owner[ID]-->' + result.InterfaceResult.Versions[j]['Owner']['ID']);
                console.log('StorageClass-->' + result.InterfaceResult.Versions[j]['StorageClass']);
            }
            console.log('DeleteMarkers:');
            for(let i=0;i<result.InterfaceResult.DeleteMarkers.length;i++){
                console.log('DeleteMarker[' + i +  ']:');
                console.log('Key-->' + result.InterfaceResult.DeleteMarkers[i]['Key']);
                console.log('VersionId-->' + result.InterfaceResult.DeleteMarkers[i]['VersionId']);
                console.log('IsLatest-->' + result.InterfaceResult.DeleteMarkers[i]['IsLatest']);
                console.log('LastModified-->' + result.InterfaceResult.DeleteMarkers[i]['LastModified']);
                console.log('Owner[ID]-->' + result.InterfaceResult.DeleteMarkers[i]['Owner']['ID']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.7 List Multipart uploads

API Description

You can use this API to list all the multipart uploads that are initialized but not combined or aborted in a specified bucket.

Method Definition

```
ObsClient.listMultipartUploads
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Delimiter	String	Optional	Character used to group object names involved in multipart uploads. If the object name contains the **Delimiter** parameter, the character string from the first character to the first **Delimiter** parameter in the object name is grouped under a single result element, **CommonPrefix**. (If a prefix is specified in the request, the prefix must be removed from the object name.)
Prefix	String	Optional	Prefix that the object names in the multipart uploads to be listed must contain
MaxUploads	Number	Optional	Maximum number of returned multipart uploads. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.
KeyMarker	String	Optional	Object name to start with when listing multipart uploads
UploadIdMarker	String	Optional	Upload ID after which the multipart upload listing begins. It is effective only when used with **KeyMarker**, so that multipart uploads after **UploadIdMarker** of **KeyMarker** will be listed.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Bucket	String	Bucket name

Field			Type	Description
KeyMarker			String	Object name after which listing multipart uploads begins, which is consistent with that set in the request
UploadIdMarker			String	Upload ID after which the multipart upload listing begins, which is consistent with that set in the request
NextKeyMarker			String	Object name to start with upon the next request for listing multipart uploads
NextUploadIdMarker			String	Upload ID to start with upon the next request for listing multipart uploads. It must be used with the **NextKeyMarker** parameter.
Delimiter			String	Character used to group object names in multipart uploads, which is consistent with that set in the request
Prefix			String	Object name prefix in multipart uploads, which is consistent with that set in the request
MaxUploads			String	Maximum number of listed multipart uploads, which is consistent with that set in the request
IsTruncated			String	Whether all multipart uploads are returned for a request. If the field value is **true**, not all multipart uploads are returned. If the field value is **false**, all multipart uploads are returned.
Uploads			Array	List of multipart uploads.
	Key		String	Name of the object to be uploaded
	UploadId		String	Multipart upload ID
	Initiator		Object	Initiator of the multipart upload
		ID	String	ID of the domain to which the owner belongs
	Owner		Object	Owner of the multipart upload. It is the same as the initiator.
		ID	String	ID of the domain to which the owner belongs
	Initiated		String	Time when the multipart upload was initiated

Field		Type	Description
	StorageClass	String	Storage class of the object to be uploaded
CommonPrefixes		Array	List of object name prefixes grouped according to the **Delimiter** parameter (if specified)
	Prefix	String	Object name prefix grouped according to the **Delimiter** parameter

Sample Code

```
obsClient.listMultipartUploads ({
    Bucket : 'bucketname',
    Prefix : 'prefix',
    MaxUploads : 100
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Bucket-->' + result.InterfaceResult.Bucket);
            for(let i=0;i<result.InterfaceResult.Uploads.length;i++){
                console.log('Uploads[' + i + ']');
                console.log('UploadId-->' + result.InterfaceResult.Uploads[i]['UploadId']);
                console.log('Key-->' + result.InterfaceResult.Uploads[i]['Key']);
                console.log('Initiated-->' + result.InterfaceResult.Uploads[i]['Initiated']);
                console.log('StorageClass-->' + result.InterfaceResult.Uploads[i]['StorageClass']);
                console.log('Owner[ID]-->' + result.InterfaceResult.Uploads[i]['Owner']['ID']);
                console.log('Initiator[ID]-->' + result.InterfaceResult.Uploads[i]['Initiator']['ID']);
            }
        }else{
            console.log('Status-->' + result.CommonMsg.Status);
        }
    }
});
```

4.8 Obtain Bucket Metadata

API Description

You can use this API to send a HEAD request to a bucket to obtain the bucket metadata such as storage class and CORS rules (if set).

Method Definition

```
ObsClient.getBucketMetadata
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Origin	String	Optional	Origin of the cross-domain request specified by the pre-request. Generally, it is a domain name.
RequestHeader	String	Optional	HTTP headers in the cross-domain request

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
StorageClass	String	Storage class of the bucket. When the storage class is OBS Standard, the value is null.
Location	String	Bucket location
ObsVersion	String	OBS server version
AllowOrigin	String	If **Origin** in the request meets the CORS rule of the server, **AllowedOrigin** in the CORS configuration of the server is returned.
AllowHeader	String	If **RequestHeader** in the request meets the CORS rule of the server, **AllowedHeader** in the CORS configuration of the server is returned.
AllowMethod	String	**AllowedMethod** in the CORS rule of the server
ExposeHeader	String	**ExposeHeader** in the CORS rule of the server
MaxAgeSeconds	String	**MaxAgeSeconds** in the CORS rule of the server

Sample Code

```
obsClient.getBucketMetadata({
    Bucket : 'bucketname',
```



```
    Origin : 'http://www.a.com',
    RequestHeader : 'x-obs-header'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
            console.log('Location-->' + result.InterfaceResult.Location);
            console.log('AllowOrigin-->' + result.InterfaceResult.AllowOrigin);
            console.log('AllowHeaders-->' + result.InterfaceResult.AllowHeader);
            console.log('AllowMethods-->' + result.InterfaceResult.AllowMethod);
            console.log('MaxAgeSeconds-->' + result.InterfaceResult.MaxAgeSeconds);
            console.log('ExposeHeaders-->' + result.InterfaceResult.ExposeHeader);
        }else{
            console.log('Status-->' + result.CommonMsg.Status);
        }
    }
});
```

4.9 GET Bucket location

API Description

You can use this API to obtain the bucket location.

Method Definition

```
ObsClient.getBucketLocation
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Location	String	Bucket location

Sample Code

```
obsClient.getBucketLocation({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
```



```
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Location-->' + result.InterfaceResult.Location);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.10 GET Bucket storageinfo

API Description

You can use this API to obtain storage information about a bucket, including the bucket size and number of objects in the bucket.

Method Definition

```
ObsClient.getBucketStorageInfo
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Size	String	Bucket size
ObjectNumber	String	Number of objects in the bucket

Sample Code

```
obsClient.getBucketStorageInfo({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Size-->' + result.InterfaceResult.Size);
            console.log('ObjectNumber-->' + result.InterfaceResult.ObjectNumber);
        }else{
```



```
                console.log('Code-->' + result.CommonMsg.Code);
                console.log('Message-->' + result.CommonMsg.Message);
            }
        }
});
```

4.11 PUT Bucket quota

API Description

You can use this API to set the bucket quota. A bucket quota must be expressed in bytes and the maximum value is 2^{63}-1. Value **0** indicates that no upper limit is set for the bucket quota.

Method Definition

```
ObsClient.setBucketQuota
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
StorageQuota	Number	Mandatory	Bucket quota. The value is a non-negative integer.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketQuota ({
    Bucket : 'bucketname',
    StorageQuota : 1024 * 1024 * 1024
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.12 GET Bucket quota

API Description

You can use this API to obtain the bucket quota. Value **0** indicates that no upper limit is set for the bucket quota.

Method Definition

```
ObsClient.getBucketQuota
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
StorageQuota	String	Bucket quota

Sample Code

```
obsClient.getBucketQuota ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('StorageQuota-->' + result.InterfaceResult.StorageQuota);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.13 Set Bucket storagePolicy

API Description

You can use this API to set storage classes for buckets. The storage class of an object defaults to be that of its residing bucket.

Method Definition

```
ObsClient.setBucketStoragePolicy
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
StorageClass	String	Mandatory	**Storage class** of the bucket

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketStoragePolicy({
    Bucket : 'bucketname',
    StorageClass: obsClient.enums.StorageClassWarm
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.14 GET Bucket storagePolicy

API Description

You can use this API to obtain the storage class of a bucket.

Method Definition

```
ObsClient.getBucketStoragePolicy
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
StorageClass	String	Storage class of the bucket

Sample Code

```
obsClient.getBucketStoragePolicy ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.15 PUT Bucket acl

API Description

You can use this API to set the ACL of a bucket.

Method Definition

```
ObsClient.setBucketAcl
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			String	Mandatory	Bucket name
ACL			String	Optional	**Pre-defined access control policy**
Owner			Object	Optional	Bucket owner
	ID		String	Mandatory	ID of the domain to which the bucket owner belongs
Grants			Array	Optional	List of grantees' permission information
	Grantee		Object	Mandatory	Grantee
		Type	String	Mandatory	**Available Grantee Types**
		ID	String	If **Type** is **CanonicalUser**, this field is mandatory. If **Type** is **Group**, this field must be **null**.	ID of the domain to which the grantee belongs

Field			Type	Optional or Mandatory	Description
		URI	String	If **Type** is **Group**, this field is mandatory. If **Type** is **CanonicalUser**, this field must be **null**.	**Available Grantee Groups**
	Permission		String	Mandatory	**Granted permission**
	Delivered		Boolean	Optional	Whether an object inherits the ACL of its residing bucket

NOTE

- **Owner** and **Grants** must be used together. These two fields are mutually exclusive with **ACL**. When **ACL** is configured, these two fields are unavailable, and vice versa.
- You must set either the two fields or **ACL**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketAcl ({
      Bucket : 'bucketname',
      Owner:{ID:'ownerid'},
      Grants:[
            {Grantee:{Type:'CanonicalUser',ID:'userid'},Permission:obsClient.enums.PermissionRead},
            {Grantee:{Type:'CanonicalUser',ID:'userid'},Permission:obsClient.enums.PermissionWrite},
            {Grantee:{Type:'Group', URI: obsClient.enums.GroupLogDelivery},Permission:
obsClient.enums.PermissionWrite},
            {Grantee:{Type:'Group', URI: obsClient.enums.GroupLogDelivery},Permission:
obsClient.enums.PermissionWriteAcp}
      ]
},(err, result) => {
      if(err){
            console.error('Error-->' + err);
```



```
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.16 GET Bucket acl

API Description

You can use this API to obtain a bucket ACL.

Method Definition

```
ObsClient.getBucketAcl
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field			Type	Description
RequestId			String	Request ID returned by the OBS server
Owner			Object	Bucket owner
	ID		String	ID of the domain to which the bucket owner belongs
	Name		String	Name of the bucket owner
Grants			Array	List of grantees' permission information
	Grantee		Object	Grantee
		Name	String	Grantee name. This field is null when **Type** of **Grantee** is **Group**.
		ID	String	ID of the domain to which the grantee belongs. This field is null when **Type** of **Grantee** is **Group**.

Field			Type	Description
		URI	String	Grantee group. This field is null when **Type** of **Grantee** is **CanonicalUser**.
	Permission		String	Granted permission
	Delivered		String	Whether an object inherits the ACL of its residing bucket

Sample Code

```
obsClient.getBucketAcl ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Owner[ID]-->' + result.InterfaceResult.Owner.ID);
            console.log('Grants:');
            for(let i=0;i<result.InterfaceResult.Grants.length;i++){
                console.log('Grant[' + i + ']:');
                console.log('Grantee[ID]-->' + result.InterfaceResult.Grants[i]['Grantee']['ID']);
                console.log('Grantee[URI]-->' + result.InterfaceResult.Grants[i]['Grantee']['URI']);
                console.log('Permission-->' + result.InterfaceResult.Grants[i]['Permission']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.17 PUT Bucket logging

API Description

You can use this API to set the access logging settings for a bucket.

Method Definition

```
ObsClient.setBucketLogging
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Field				Type	Optional or Mandatory	Description
Agency				String	Mandatory when configuring bucket logging	Agency name
LoggingEnabled				Object	Optional	Log configuration information
	TargetBucket			String	Mandatory	Target bucket for which logs are generated
	TargetPrefix			String	Mandatory	Name prefix of a to-be-logged object in the target bucket
	TargetGrants			Array	Optional	List of grantees' permission information
		Grantee		Object	Optional	Grantee
			Type	String	Mandatory	**Grantee type**
			ID	String	If **Type** is **Canonical User**, this field is mandatory. If **Type** is **Group**, this field must be **null**.	ID of the domain to which the grantee belongs
			URI	String	If **Type** is **Group**, this field is mandatory. If **Type** is **Canonical User**, this field must be **null**.	**Grantee group**
		Permission		String	Optional	**Granted permission**

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketLogging ({
	Bucket : 'bucketname',
	Agency : 'your agency',
	LoggingEnabled:{
		TargetBucket:'targetbucketname',
		TargetPrefix:'prefix',
		TargetGrants:[
			{Grantee: {Type:'CanonicalUser',ID:'userid'},Permission: obsClient.enums.PermissionRead},
			{Grantee: {Type:'Group',URI: obsClient.enums.GroupAllUsers},Permission:
obsClient.enums.PermissionRead}
		]
	}
},(err, result) => {
	if(err){
		console.error('Error-->' + err);
	}else{
		if(result.CommonMsg.Status < 300){
			console.log('RequestId-->' + result.InterfaceResult.RequestId);
		}else{
			console.log('Code-->' + result.CommonMsg.Code);
			console.log('Message-->' + result.CommonMsg.Message);
		}
	}
});
```

4.18 GET Bucket logging

API Description

You can use this API to obtain the access logging settings of a bucket.

Method Definition

```
ObsClient.getBucketLogging
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field				Type	Description
RequestId				String	Request ID returned by the OBS server
Agency				String	Agency name
LoggingEnabled				Object	Log configuration information
	TargetBucket			String	Target bucket for which logs are generated
	TargetPrefix			String	Name prefix of a to-be-logged object in the target bucket
	TargetGrants			Array	List of grantees' permission information
		Grantee		Object	Grantee
			ID	String	ID of the domain to which the grantee belongs. This field is null when **Type** of **Grantee** is **Group**.
			URI	String	Grantee group. This field is null when **Type** of **Grantee** is **CanonicalUser**.
		Permission		String	Granted permission

Sample Code

```
obsClient.getBucketLogging ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            if(result.InterfaceResult.LoggingEnabled){
                console.log('TargetBucket-->' + result.InterfaceResult.LoggingEnabled.TargetBucket);
                console.log('TargetPrefix-->' + result.InterfaceResult.LoggingEnabled.TargetPrefix);
                for(let i=0;i<result.InterfaceResult.LoggingEnabled.TargetGrants.length;i++){
                    console.log('Grant[' + i + ']:');
                    console.log('Grantee[ID]-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Grantee']['ID']);
                    console.log('Grantee[URI]-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Grantee']['URI']);
                    console.log('Permission-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Permission']);
                }
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.19 PUT Bucket policy

API Description

You can use this API to set the bucket policy. If the bucket already has a policy, the policy will be overwritten by the one specified in this request.

Method Definition

```
ObsClient.setBucketPolicy
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Policy	String	Mandatory	Policy information in the JSON format. For details, see **Bucket Policy Parameters**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketPolicy ({
    Bucket : 'bucketname',
    Policy: 'your policy',
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


NOTICE

The bucket name contained in the **Resource** field in **Policy** must be the one specified for the bucket policy.

4.20 GET Bucket policy

API Description

You can use this API to obtain the bucket policy.

Method Definition

```
ObsClient.getBucketPolicy
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Policy	String	Policy information in the JSON format

Sample Code

```
obsClient.getBucketPolicy ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Policy-->' + result.InterfaceResult.Policy);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.21 DELETE Bucket policy

API Description

You can use this API to delete a bucket policy.

Method Definition

```
ObsClient.deleteBucketPolicy
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucketPolicy ({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.22 PUT Bucket lifecycle

API Description

You can use this API to set lifecycle rules for a bucket, so as to periodically transit storage classes of objects and delete objects in the bucket.

Method Definition

```
ObsClient.setBucketLifecycle
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			String	Mandatory	Bucket name
Rules			Array	Mandatory	Lifecycle rules of the bucket
	Transitions		Array	Optional	List of object transition policies
		StorageClass	String	Mandatory	**Storage class** of the object after transition **NOTE** The Standard storage class is not supported.
		Date	String	This parameter is mandatory if **Days** is not set.	Date when an object will be transited The value must conform with the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01T00:00:00Z.

Field			Type	Optional or Mandatory	Description
		Days	Number	This parameter is mandatory if **Date** is not set.	Number of days after which an object will be transited since its creation. The value must be a positive integer.
	Expiration		Object	Optional	Expiration time of an object
		Date	String	This parameter is mandatory if **Days** is not set.	Date when an object expires. The value must conform with the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01T00:00:00Z.
		Days	Number	This parameter is mandatory if **Date** is not set.	Number of days after which an object expires since its creation. The parameter value is a positive integer.

Field		Type	Optional or Mandatory	Description
	ID	String	Optional	Rule ID. The value is a string of no more than 255 characters.
	Prefix	String	Mandatory	Object name prefix. It identifies the objects to which the rule applies. The value can be empty, indicating that the rule applies to all objects in the bucket.
	Status	String	Mandatory	Whether the current rule is enabled. Possible values are: • Enabled • Disabled
	NoncurrentVersionTransitions	Array	Optional	List of noncurrent object version transition policies

Field			Type	Optional or Mandatory	Description
		StorageClass	String	Mandatory	**Storage class** of the noncurrent object version after transition
		NoncurrentDays	Number	Mandatory	Number of days after which an object will be transited since it becomes a noncurrent version. The parameter value must be a positive integer.
	NoncurrentVersionExpiration		Object	Optional	Expiration time of a noncurrent object version
		NoncurrentDays	Number	Mandatory	Number of days after which an object expires since it becomes a noncurrent version. The parameter value must be a positive integer.

NOTE

Transition, **Expiration**, **NoncurrentVersionTransitions**, and **NoncurrentVersionExpiration** cannot be all null.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketLifecycle({
    Bucket : 'bucketname',
    Rules:[
        {ID:'rule1',Prefix:'prefix1',Status:'Enabled',Expiration:{Days: 60}, NoncurrentVersionExpiration:
{NoncurrentDays : 60}},
        {ID:'rule2',Prefix:'prefix2',Status:'Enabled',Expiration:{Date: '2018-12-31T00:00:00Z'}}
    ]
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.23 GET Bucket lifecycle

API Description

You can use this API to obtain the lifecycle rules of a bucket.

Method Definition

```
ObsClient.getBucketLifecycle
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field			Type	Description
RequestId			String	Request ID returned by the OBS server
Rules			Array	Lifecycle rules of the bucket
	Transitions		Array	List of object transition policies
		Storage Class	String	Storage class of the object after transition
		Date	String	Date when an object will be transited
		Days	String	Number of days after which an object will be transited since its creation
	Expiration		Object	Expiration time of an object
		Date	String	Date when an object expires
		Days	String	Number of days after which an object expires since its creation
	ID		String	Rule ID
	Prefix		String	Object name prefix. It identifies the objects to which the rule applies.
	Status		String	Whether the rule is enabled
	NoncurrentVersionTransitions		Array	List of noncurrent object version transition policies
		Storage Class	String	Storage class of the noncurrent object version after transition
		NoncurrentDays	String	Number of days after which an object will be transited since it becomes a noncurrent version
	NoncurrentVersionExpiration		Object	Expiration time of a noncurrent object version
		NoncurrentDays	String	Number of days after which an object expires since it becomes a noncurrent version.

Sample Code

```
obsClient.getBucketLifecycle({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            for(let i=0;i<result.InterfaceResult.Rules.length;i++){
```



```
                    console.log('Rule[' + i + ']:');
                    console.log('ID-->' + result.InterfaceResult.Rules[i]['ID']);
                    console.log('Prefix-->' + result.InterfaceResult.Rules[i]['Prefix']);
                    console.log('Status-->' + result.InterfaceResult.Rules[i]['Status']);
                    console.log('Expiration[Date]-->' + result.InterfaceResult.Rules[i]['Expiration']['Date']);
                    console.log('Expiration[Days]-->' + result.InterfaceResult.Rules[i]['Expiration']['Days']);
                    console.log('NoncurrentVersionExpiration[NoncurrentDays]-->' +
result.InterfaceResult.Rules[i]['NoncurrentVersionExpiration']['NoncurrentDays']);
                }
            }else{
                console.log('Code-->' + result.CommonMsg.Code);
                console.log('Message-->' + result.CommonMsg.Message);
            }
        }
});
```

4.24 DELETE Bucket lifecycle

API Description

You can use this API to delete all lifecycle rules of a bucket.

Method Definition

```
ObsClient.deleteBucketLifecycle
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucketLifecycle({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
```



```
    }
});
```

4.25 PUT Bucket website

API Description

You can use this API to set website hosting for a bucket.

Method Definition

```
ObsClient.setBucketWebsite
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
RedirectAllRequestsTo		Object	Optional	Redirection rule of all requests
	HostName	String	Mandatory	Host name used for redirection
	Protocol	String	Optional	Protocol used for redirection. Possible values are: • **http** (default) • https
ErrorDocument		Object	Optional	Error page settings
	Key	String	Optional	Page to return to when a 4*XX* error occurs
IndexDocument		Object	Optional	Default page settings

Field			Type	Optional or Mandatory	Description
	Suffix		String	Mandatory	Suffix that is appended to a request initiated for a directory on the website endpoint. For example, if the suffix is **index.html** and you request for **samplebucket/images/**, the data that is returned will be the object with the key name **images/index.html** in the **samplebucket** bucket. This field cannot be left blank or contain a slash (/).
RoutingRules			Array	Optional	Redirection rule list
	Condition		Object	Optional	Matching conditions of a redirection rule
		HttpErrorCodeReturnedEquals	String	Optional	HTTP error code to be matched when a redirection rule takes effect
		KeyPrefixEquals	String	Optional	Object name prefix to be matched when a redirection rule takes effect
	Redirect		Object	Mandatory	Details about a redirection request
		Protocol	String	Optional	Protocol used for redirection. Possible values are: • http • https
		HostName	String	Optional	Host name used for redirection
		ReplaceKeyPrefixWith	String	Optional	Object name prefix used in the redirection request
		ReplaceKeyWith	String	Optional	Object name used in the redirection request. This parameter cannot be used together with **ReplaceKeyPrefixWith**.
		HttpRedirectCode	String	Optional	HTTP status code in the response to the redirection request

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 null.
- You must set either these three fields or **RedirectAllRequestsTo**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketWebsite({
        Bucket : 'bucketname',
//      RedirectAllRequestsTo : {HostName : 'www.example.com', Protocol : 'https'}
        IndexDocument:{Suffix:'index.html'},
        ErrorDocument:{Key:'error.html'},
        RoutingRules:[
            {Condition:{HttpErrorCodeReturnedEquals:'404'},Redirect:
{Protocol:'http',ReplaceKeyWith:'NotFound.html'}},
            {Condition:{HttpErrorCodeReturnedEquals:'404'},Redirect:
{Protocol:'https',ReplaceKeyWith:'test.html'}}
        ]
},(err, result) => {
        if(err){
            console.error('Error-->' + err);
        }else{
            if(result.CommonMsg.Status < 300){
                console.log('RequestId-->' + result.InterfaceResult.RequestId);
            }else{
                console.log('Code-->' + result.CommonMsg.Code);
                console.log('Message-->' + result.CommonMsg.Message);
            }
        }
});
```

4.26 GET Bucket website

API Description

You can use this API to obtain the website hosting settings of a bucket.

Method Definition

```
ObsClient.getBucketWebsite
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field			Type	Description
RequestId			String	Request ID returned by the OBS server
RedirectAllRequestsTo			Object	Redirection rule of all requests
	HostName		String	Host name used for redirection
	Protocol		String	Host name used for redirection
ErrorDocument			Object	Error page settings
	Key		String	Page to return to when a *4XX* error occurs
IndexDocument			Object	Default page settings
	Suffix		String	Suffix that is appended to a request initiated for a directory on the website endpoint. For example, if the suffix is **index.html** and you request for **samplebucket/images/**, the data that is returned will be the object with the key name **images/index.html** in the **samplebucket** bucket. This field cannot be left blank or contain a slash (/).
RoutingRules			Array	Redirection rule list
	Condition		Object	Matching conditions of a redirection rule
		HttpErrorCodeReturnedEquals	String	HTTP error code to be matched when a redirection rule takes effect
		KeyPrefixEquals	String	Object name prefix to be matched when a redirection rule takes effect
	Redirect		Object	Details about a redirection request

Field			Type	Description
		Protocol	String	Protocol used for redirection
		HostName	String	Host name used for redirection
		ReplaceKeyPrefixWith	String	Object name prefix used in the redirection request
		ReplaceKeyWith	String	Object name used in the redirection request. This parameter cannot be used together with **ReplaceKeyPrefixWith**.
		HttpRedirectCode	String	HTTP status code in the response to the redirection request

Sample Code

```
obsClient.getBucketWebsite({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('RedirectAllRequestsTo:');
            console.log('HostName-->' + result.InterfaceResult.RedirectAllRequestsTo['HostName']);
            console.log('Protocol-->' + result.InterfaceResult.RedirectAllRequestsTo['Protocol']);
            console.log('IndexDocument[Suffix]-->' + result.InterfaceResult.IndexDocument['Suffix']);
            console.log('ErrorDocument[Key]-->' + result.InterfaceResult.ErrorDocument['Key']);
            console.log('RoutingRules:');
            for(let i=0;i<result.InterfaceResult.RoutingRules.length;i++){
                console.log('RoutingRule[' + i + ']:');
                let RoutingRule = result.InterfaceResult.RoutingRules[i];
                console.log('Condition[HttpErrorCodeReturnedEquals]-->' + RoutingRule['Condition']
['HttpErrorCodeReturnedEquals']);
                console.log('Condition[KeyPrefixEquals]-->' + RoutingRule['Condition']['KeyPrefixEquals']);
                console.log('Redirect[HostName]-->' + RoutingRule['Redirect']['HostName']);
                console.log('Redirect[HttpRedirectCode]-->' + RoutingRule['Redirect']
['HttpRedirectCode']);
                console.log('Redirect[Protocol]-->' + RoutingRule['Redirect']['Protocol']);
                console.log('Redirect[ReplaceKeyPrefixWith]-->' + RoutingRule['Redirect']
['ReplaceKeyPrefixWith']);
                console.log('Redirect[ReplaceKeyWith]-->' + RoutingRule['Redirect']['ReplaceKeyWith']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.27 DELETE Bucket website

API Description

You can use this API to delete the website hosting settings of a bucket.

Method Definition

```
ObsClient.deleteBucketWebsite
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucketWebsite({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.28 PUT Bucket versioning

API Description

You can use this API to set the versioning status for a bucket.

Method Definition

```
ObsClient.setBucketVersioning
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
VersionStatus	String	Mandatory	Versioning status of the bucket. Possible values are: • Enabled • Suspended

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketVersioning({
    Bucket : 'bucketname',
    VersionStatus : 'Enabled'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.29 GET Bucket versioning

API Description

You can use this API to obtain the versioning status of a bucket.

Method Definition

```
ObsClient.getBucketVersioning
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
VersionStatus	String	Versioning status of the bucket

Sample Code

```
obsClient.getBucketVersioning({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('VersionStatus-->' + result.InterfaceResult.VersionStatus);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.30 PUT Bucket cors

API Description

You can use this API to set CORS rules for a bucket to allow the client browsers to send cross-domain requests.

Method Definition

```
ObsClient.setBucketCors
```


Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
CorsRules		Array	Mandatory	CORS rules of the bucket
	ID	String	Optional	CORS rule ID. The value is a string of no more than 255 characters.
	AllowedMethod	Array of Strings	Mandatory	HTTP methods allowed by the CORS rule. Possible values are: • GET • PUT • HEAD • POST • DELETE
	AllowedOrigin	Array of Strings	Mandatory	Origins (character strings representing domain names) allowed by the CORS rule. The value can contain one wildcard character (*). Each **AllowedOrigin** can only contain one or zero wildcard character (*).
	AllowedHeader	Array of Strings	Optional	Request headers allowed by the CORS rule. The value must not contain spaces. The value can contain one wildcard character (*). Each **AllowedHeader** can only contain one or zero wildcard character (*).
	MaxAgeSeconds	Number	Optional	Cache duration (in seconds) of the cross-region request result in the client allowed by the CORS rule. The value must be an integer.
	ExposeHeader	Array of Strings	Optional	Additional response headers allowed by the CORS rule. The value must not contain spaces.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketCors({
    Bucket : 'bucketname',
    CorsRules:[
        {
            ID:'rule1',
            AllowedMethod:['PUT','POST','GET','DELETE','HEAD'],
            AllowedOrigin:['obs.hostname','obs.hostname1'],
            AllowedHeader:['obs-header-1'],
            MaxAgeSeconds:60
        },
        {
            ID:'rule2',
            AllowedMethod:['PUT','POST','GET'],
            AllowedOrigin:['obs.hostname','obs.hostname1'],
            AllowedHeader:['header-1','header-2'],
            MaxAgeSeconds:50
        }
    ]
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.31 GET Bucket cors

API Description

You can use this API to obtain the CORS rules of a specified bucket.

Method Definition

```
ObsClient.getBucketCors
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
CorsRules		Array	CORS rules of the bucket
	ID	String	CORS rule ID
	AllowedMethod	Array of Strings	HTTP methods allowed by the CORS rule
	AllowedOrigin	Array of Strings	Origins (character strings representing domain names) allowed by the CORS rule
	AllowedHeader	Array of Strings	Request headers allowed by the CORS rule
	MaxAgeSeconds	String	Cache duration (in seconds) of the cross-region request result in the client allowed by the CORS rule.
	ExposeHeader	Array of Strings	Additional response headers allowed by the CORS rule

Sample Code

```
obsClient.getBucketCors({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            for(let k=0;k<result.InterfaceResult.CorsRules.length;k++){
                console.log('CorsRule[',k,']');
                console.log('CorsRule[ID]-->' + result.InterfaceResult.CorsRules[k]['ID']);
                console.log('CorsRule[MaxAgeSeconds]-->' + result.InterfaceResult.CorsRules[k]
['MaxAgeSeconds']);
                for (let i=0;i<result.InterfaceResult.CorsRules[k]['AllowedMethod'].length;i++){
                    console.log('CorsRule[AllowedMethod][' , i ,']-->'+result.InterfaceResult.CorsRules[k]
['AllowedMethod'][i]);
                }
                for(let i=0;i< result.InterfaceResult.CorsRules[k]['AllowedOrigin'].length;i++){
                    console.log('CorsRule[AllowedOrigin][',i ,']-->'+result.InterfaceResult.CorsRules[k]
['AllowedOrigin'][i]);
                }
                for(let i=0;i<result.InterfaceResult.CorsRules[k]['AllowedHeader'].length;i++){
                    console.log('CorsRule[AllowedHeader][',i,']-->'+result.InterfaceResult.CorsRules[k]
['AllowedHeader'][i]);
                }
                for(let i=0;i<result.InterfaceResult.CorsRules[k]['ExposeHeader'].length;i++){
                    console.log('CorsRule[ExposeHeader][',i ,']-->'+result.InterfaceResult.CorsRules[k]
['ExposeHeader'][i]);
                }
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
```



```
        }
    }
});
```

4.32 DELETE Bucket cors

API Description

You can use this API to delete the CORS rules of a specified bucket.

Method Definition

```
ObsClient.deleteBucketCors
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucketCors({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

4.33 PUT Bucket tagging

API Description

You can use this API to set bucket tags.

Method Definition

```
ObsClient.setBucketTagging
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
Tags		Array	Mandatory	Bucket tag set
	Key	String	Mandatory	Tag name, which contains 1 to 36 characters and cannot include non-printable ASCII characters (0–31) and the following special characters: *<>\= The tag keys in one bucket must be unique.
	Value	String	Mandatory	Tag value, which can contain up to 43 characters and cannot include non-printable ASCII characters (0–31) and the following special characters: *<>\=

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setBucketTagging({
    Bucket : 'bucketname',
    Tags : [{Key:'tag1', Value : 'value1'}, {Key:'tag2', Value : 'value2'}]
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.34 GET Bucket tagging

API Description

You can use this API to obtain the tags of a specified bucket.

Method Definition

```
ObsClient.getBucketTagging
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
Tags		Array	Bucket tag set
	Key	String	Tag name, which contains 1 to 36 characters
	Value	String	Tag value, which can contain up to 43 characters

Sample Code

```
obsClient.getBucketTagging({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            result.InterfaceResult.Tags.forEach(function(tag){
                console.log('Tag-->' + tag.Key + ':' + tag.Value);
            });
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


4.35 DELETE Bucket tagging

API Description

You can use this API to delete the tags of a specified bucket.

Method Definition

```
ObsClient.deleteBucketTagging
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.deleteBucketTagging({
    Bucket : 'bucketname'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


5 Object-Related APIs

5.1 PUT Object

API Description

You can use this API to upload an object to a specified bucket.

NOTE

Object-Related API functions of **ObsClient** are case insensitive. For example, **ObsClient.putObject** and **ObsClient.PutObject** indicate the same function.

Method Definition

```
ObsClient.putObject
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
ACL	String	Optional	**Pre-defined access control policy** specified for the object
StorageClass	String	Optional	**Storage class** specified for the object

Field	Type	Optional or Mandatory	Description
Body	String or stream.Readable	Optional	Content of the object to be uploaded. Both character strings and instances of **stream.Readable** are supported.
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. This field is effective when **SourceFile** is set and its default value is **0**.
SourceFile	String	Optional	Path to the source file of the object
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress **NOTE** This callback function contains the following parameters in sequence: Number of uploaded bytes, total bytes, and used time (unit: second).
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
Expires	Number	Optional	Expiration time of the object, in days
SuccessActionRe-direct	String	Optional	Redirection address after the upload is successful
ContentType	String	Optional	MIME type of the object
ContentLength	Number	Optional	Object length. This field is effective when **SourceFile** is set.
ContentMD5	String	Optional	MD5 value of the object (Base64-encoded). It is provided for the OBS server to verify data integrity.
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: • kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be **null**.

Field	Type	Optional or Mandatory	Description
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	Buffer	Optional	Key used to encrypt the object in SSE-C mode, which is calculated by using AES256

NOTE

- **Body** and **SourceFile** cannot be used together.
- If both **Body** and **SourceFile** are **null**, the size of the uploaded object is 0 bytes.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	Object ETag
VersionId	String	Object version ID
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is null.
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.putObject({
        Bucket : 'bucketname',
        Key : 'objectkey',
        Metadata:{meta1:'value1', meta2:'value2'},
//      SourceFile : 'localfile',
        Body : 'Hello OBS',
        ContentType: 'text/plain'
},(err, result) => {
        if(err){
                console.error('Error-->' + err);
        }else{
```



```
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('ETag-->' + result.InterfaceResult.ETag);
            console.log('VersionId-->' + result.InterfaceResult.VersionId);
            console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.2 Append Object

API Description

You can use this API to upload an object in appendable mode and append data to the object.

Method Definition

```
ObsClient.appendObject
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
Position	Number	Mandatory	Start position for next appending. This parameter must be set to **0** when you create an appendable object.
ACL	String	Optional	**Pre-defined access control policy** specified during the object creation
StorageClass	String	Optional	**Storage class** specified during the object creation
Body	String or stream.Readable	Optional	Content of the object to be uploaded. Both character strings and instances of **stream.Readable** are supported.

Field	Type	Optional or Mandatory	Description
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. This field is effective when **SourceFile** is set and its default value is **0**.
SourceFile	String	Optional	Path to the source file of the object
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
Expires	Number	Optional	Expiration time of the object, in days
SuccessActionRe-direct	String	Optional	Redirection address after the upload is successful
ContentType	String	Optional	MIME type of the object
ContentLength	Number	Optional	Object length. This field is effective when **SourceFile** is set.
ContentMD5	String	Optional	MD5 value of the object (Base64-encoded). It is provided for the OBS server to verify data integrity.
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: • kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be **null**.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	Buffer	Optional	Key used to encrypt the object in SSE-C mode, which is calculated by using AES256

NOTE

- **Body** and **SourceFile** cannot be used together.
- If both **Body** and **SourceFile** are **null**, the size of the uploaded object is **0** bytes.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	Object ETag
NextPosition	String	Start position for next appending
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is **null**.
SseKms	String	Algorithm used in SSE-KMS decryption
SseKmsKey	String	Key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C decryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C decryption

Sample Code

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

// Create an instance of ObsClient.
var obsClient = new ObsClient({
    //Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
    //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
    access_key_id: process.env.ACCESS_KEY_ID,
    secret_access_key: process.env.SECRET_ACCESS_KEY,
    server : 'https://your-endpoint'
});

// Create an appendable object. The start position must be 0.
obsClient.appendObject({
    Bucket:'bucketname',
    Key:'objectname',
    Position : 0,
    Body : 'Hello OBS'
}).then(function(result){
    console.log('Status-->' + result.CommonMsg.Status);
    if(result.CommonMsg.Status < 300 && result.InterfaceResult){
        console.log('NextPosition-->' + result.InterfaceResult.NextPosition);
    }
    // Append data to the object.
    obsClient.appendObject({
        Bucket:'bucketname',
```

```
        Key:'objectname',
        Position : result.InterfaceResult.NextPosition,
        Body : 'Hello OBS Again'
    }, function(err, result2){
        if(err){
            console.error('Error-->' + err);
        }else{
            console.log('Status-->' + result2.CommonMsg.Status);
            if(result2.CommonMsg.Status < 300 && result2.InterfaceResult){
                console.log('NextPosition-->' + result2.InterfaceResult.NextPosition);
            }
        }
    });

    // Use the API for obtaining object properties to get the start position for next appending.
    obsClient.getObjectMetadata({
        Bucket:'bucketname',
        Key:'objectname',
    }).then(function(result3){
        console.log('Status-->' + result3.CommonMsg.Status);
        if(result3.CommonMsg.Status < 300 && result3.InterfaceResult){
            console.log('RequestId-->' + result3.InterfaceResult.RequestId);
            console.log('NextPosition-->' + result3.InterfaceResult.NextPosition);
        }
    }).catch(function(err){
        console.error('err:' + err);
    });

}).catch(function(err){
    console.error('err:' + err);
});
```

NOTE

- Use the **Position** parameter to specify the start position for next appending and set it to **0** when you create an appendable object.
- Objects uploaded using **ObsClient.putObject**, referred to as normal objects, can overwrite objects uploaded using **ObsClient.appendObject**, referred to as appendable objects. Data cannot be appended to an appendable object anymore once the object has been overwritten by a normal object.
- When you upload an object for the first time in appendable mode, an exception will be thrown (status code **409**) if a normal object with the same name exists.
- The ETag returned for an appendable upload is the ETag for the uploaded content, rather than that of the whole object.
- Data size in each appendable upload cannot exceed 5 GB, and 10,000 times of appendable uploads can be performed on a single object.
- After an appendable upload is complete successfully, you can use **InterfaceResult.NextPosition** obtained from the returned result or call **ObsClient.getObjectMetadata**, to get the location for next appending.

5.3 GET Object

API Description

You can user this API to download an object in a specified bucket.

Method

```
ObsClient.getObject
```


Request Parameters

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID
ProgressCallback	Function	Optional	Callback function for obtaining the download progress **NOTE** This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second).
IfMatch	String	Optional	Returns the source object if its ETag is the same as the one specified by this parameter; otherwise, an error code is returned.
IfModifiedSince	String	Optional	Returns the object if it is modified after the time specified by this parameter; otherwise, an error code is returned. This parameter must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt.
IfNoneMatch	String	Optional	Returns the source object if its ETag is different from the one specified by this parameter; otherwise, an error code is returned.
IfUnmodifiedSince	String	Optional	Returns the object if it remains unchanged since the time specified by this parameter; otherwise, an error code is returned. This parameter must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt.

Field	Type	Optional or Mandatory	Description
Range	String	Optional	Download range. The value range is [0, object length-1] and is in the format of **bytes=***x-y*. The maximum length of **Range** is the length of the object minus 1. If it exceeds this value, the length of the object minus 1 is used.
Origin	String	Optional	Origin of the cross-domain request specified by the pre-request. Generally, it is a domain name.
RequestHeader	String	Optional	HTTP headers in the cross-domain request
ResponseCacheControl	String	Optional	Rewrites the **Cache-Control** header in the response.
ResponseContent-Disposition	String	Optional	Rewrites the **Content-Disposition** header in the response.
ResponseContentEncoding	String	Optional	Rewrites the **Content-Encoding** header in the response.
ResponseContentLanguage	String	Optional	Rewrites the **Content-Language** header in the response.
ResponseContent-Type	String	Optional	Rewrites the **Content-Type** header in the response.
ResponseExpires	String	Optional	Rewrites the **Expires** header in the response
SaveAsFile	String	Optional	The download path with the file name contained
SaveAsStream	Boolean	Optional	Whether the object is returned as a readable stream
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: ● AES256
SseCKey	Buffer	Optional	Key used in SSE-C decryption, which is calculated by using AES256.

NOTE

- **SaveAsFile** and **SaveAsStream** cannot be used together.
- If the download request includes **IfUnmodifiedSince** or **IfMatch**, and **IfUnmodifiedSince** or **IfMatch** is not met, **412 Precondition Failed** will be returned.
- If the download request includes **IfModifiedSince** or **IfNoneMatch**, and **IfModifiedSince** or **IfNoneMatch** is not met, **304 Not Modified** will be returned.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
DeleteMarker	String	Whether the deleted object is a delete marker
LastModified	String	Time when the last modification was made to the object
ContentLength	String	Object size in bytes
CacheControl	String	**Cache-Control** header in the response
ContentDisposition	String	**Content-Disposition** header in the response
ContentEncoding	String	**Content-Encoding** header in the response
ContentLanguage	String	**Content-Language** header in the response
ContentType	String	MIME type of the object
Expires	String	**Expires** header in the response
ETag	String	Object ETag
VersionId	String	Object version ID
WebsiteRedirectLocation	String	Location where the object is redirected to, when the bucket is configured with website hosting
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is null.
Restore	String	Restore status of the object in the OBS Archive storage class
AllowOrigin	String	If **Origin** in the request meets the CORS rules of the bucket, **AllowedOrigin** in the CORS rules is returned.

Field	Type	Description
AllowHeader	String	If **RequestHeader** in the request meets the CORS rules of the bucket, **AllowedHeader** in the CORS rules is returned.
AllowMethod	String	**AllowedMethod** in the CORS rules of the bucket
ExposeHeader	String	**ExposeHeader** in the CORS rules of the bucket
MaxAgeSeconds	String	**MaxAgeSeconds** in the CORS rules of the bucket
SseKms	String	Algorithm used in SSE-KMS decryption
SseKmsKey	String	Master key used in SSE-KMS decryption
SseC	String	Algorithm used in SSE-C decryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C decryption
Expiration	String	Expiration details
Content	String or stream.Readable	Content of the object. The content is **null** when **SaveAsFile** is set. The content is an instance of **stream.Readable** when **SaveAsStream** is set to **true**. The content is an instance of **Buffer** when neither **SaveAsFile** nor **SaveAsStream** is set.
Metadata	Object	Customized metadata of the object

Sample Code

This example downloads part (length range: 0-1,023 bytes) of **objectname** from **examplebucket** and prints the progress.

```
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');

// Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
// Obtain an AK/SK pair on the management console by referring to https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
const AK = process.env.AccessKeyID
const SK = process.env.SecretAccessKey
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, obtain them using environment variables.
const security_token= process.env.SecurityToken
// Set server to the endpoint corresponding to the bucket. EU-Dublin is used here as an example. Replace it with the one in your actual situation.
const server = "https://obs.eu-west-101.myhuaweicloud.com"

var obsClient = new ObsClient({
```



```
    access_key_id: AK,
    secret_access_key: SK,
    server: server,
});
const Bucket = 'examplebucket'
// Name of the object to be downloaded
const Key = 'objectname'
// Specify the object range to download in the format of bytes=x-y. The range must be from 0 to the total bytes of the object minus 1. If the specified range is beyond the allowed upper limit, the upper limit is used.
const Range = 'bytes=0-1023'
// Specify the download path with the file name contained.
const SaveAsFile = 'D:\\example'
try {
    obsClient.getObject({
        Bucket,
        Key,
        Range,
        SaveAsFile,
        // Progress callback
        ProgressCallback: function (transferredAmount, totalAmount, totalSeconds) {
            // Print the download speed in KB/s.
            console.log(transferredAmount * 1.0 / totalSeconds / 1024);
            // Print the upload percentage.
            console.log(transferredAmount * 100.0 / totalAmount);
        },
        // Specify whether to return the object as a readable stream.
        // SaveAsStream: true
    }, (err, result) => {
        if (err) {
            console.log('GetObject Failed')
            console.error('Error-->' + err);
        } else {
            if (result.CommonMsg.Status < 300) {
                console.log('GetObject Succeeded')
                console.log('RequestId-->' + result.InterfaceResult.RequestId);
                console.log('ETag-->' + result.InterfaceResult.ETag);
                // The following parameters are exclusive to versioned objects.
                console.log('VersionId-->' + result.InterfaceResult.VersionId);
                console.log('ContentLength-->' + result.InterfaceResult.ContentLength);
                // The following parameters are exclusive to versioned objects.
                console.log('DeleteMarker-->' + result.InterfaceResult.DeleteMarker);
                console.log('LastModified-->' + result.InterfaceResult.LastModified);
                // Object storage class. If the storage class is Standard, this parameter is left blank.
                console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
                // console.log('Content-->' + result.InterfaceResult.Content.toString());
                console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
            } else {
                console.log('GetObject Failed')
                console.log('ErrorCode-->' + result.CommonMsg.Code);
                console.log('ErrorMessage-->' + result.CommonMsg.Message);
            }
        }
    });
} catch (error) {
    console.log('GetObject Failed')
    console.error('Error-->' + error);
}
```

5.4 PUT Object - Copy

API Description

You can use this API to create a copy for an object in a specified bucket.

Method Definition

```
ObsClient.copyObject
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Target bucket name
Key	String	Mandatory	Target object name
ACL	String	Optional	**Pre-defined access control policy** specified during object copy
StorageClass	String	Optional	**Storage class** of the object. Possible values are:
CopySource	String	Mandatory	Parameter used to specify the source bucket, source object, and source object version ID which can be null. It is in the format of *SourceBucketName*/*SourceObjectName*?*versionId*=*SourceObjectVersionId*.
CopySourceIfMatch	String	Optional	Copies the source object if its ETag is the same as the one specified by this parameter; otherwise, an error code is returned.
CopySourceIfModifiedSince	String	Optional	Copies the source object if it is changed after the time specified by this parameter; otherwise, an error code is returned. This parameter must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt.
CopySourceIfNoneMatch	String	Optional	Copies the source object if its ETag is different from the one specified by this parameter; otherwise, an error code is returned.
CopySourceIfUnmodifiedSince	String	Optional	Copies the source object if it is changed before the time specified by this parameter; otherwise, an error code is returned. This parameter must conform with the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt.

Field	Type	Optional or Mandatory	Description
CacheControl	String	Optional	When an object is copied, this parameter rewrites the **Cache-Control** header in the response.
ContentDisposition	String	Optional	When an object is copied, this parameter rewrites the **Content-Disposition** header in the response.
ContentEncoding	String	Optional	When an object is copied, this parameter rewrites the **Content-Encoding** header in the response.
ContentLanguage	String	Optional	When an object is copied, this parameter rewrites the **Content-Language** header in the response.
ContentType	String	Optional	When an object is copied, this parameter rewrites the **Content-Type** header in the response.
Expires	String	Optional	When an object is copied, this parameter rewrites the **Expires** header in the response.
MetadataDirective	String	Optional	**Replication policy**
Metadata	Object	Optional	Customized metadata of the target object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
SuccessActionRe-direct	String	Optional	Redirection address after the copy is successful
SseKms	String	Optional	Algorithm used to encrypt a target object in SSE-KMS mode. The value can be: • kms
SseKmsKey	String	Optional	Master key used to encrypt a target object in SSE-KMS encryption, which can be left **null**
SseC	String	Optional	Algorithm used to encrypt a target object in SSE-C mode. The value can be: • AES256

Field	Type	Optional or Mandatory	Description
SseCKey	Buffer	Optional	Key used to encrypt a target object in SSE-C mode, which is calculated by using the AES256 algorithm.
CopySourceSseC	String	Optional	Algorithm used to decrypt a source object in SSE-C mode. The value can be: • AES256
CopySourceSseCKey	Buffer	Optional	Key used to decrypt a source object in SSE-C mode, which is calculated by using the AES256 algorithm.

NOTE

- If **CopySourceIfUnmodifiedSince**, **CopySourceIfMatch**, **CopySourceIfModifiedSince**, or **CopySourceIfNoneMatch** is included and its specified condition is not met, **412 Precondition Failed** will be returned.
- **CopySourceIfModifiedSince** and **CopySourceIfNoneMatch** can be used together. So do **CopySourceIfUnmodifiedSince** and **CopySourceIfMatch**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag of the target object
LastModified	String	Time when the last modification was made to the target object
VersionId	String	Version ID of the target object. This field is null if versioning is not enabled for the target bucket.
CopySourceVersionId	String	Version ID of the source object. This field is null if versioning is not enabled for the source bucket.
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Master key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption

Field	Type	Description
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.copyObject({
    Bucket:'bucketname',
    Key:'objectkey',
    CopySource:'srcbucketname/srcobjectkey',
    Metadata:{meta1:'value1'}
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('ETag-->' + result.InterfaceResult.ETag);
            console.log('VersionId-->' + result.InterfaceResult.VersionId);
            console.log('CopySourceVersionId-->' + result.InterfaceResult.CopySourceVersionId);
            console.log('LastModified-->' + result.InterfaceResult.LastModified);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.5 DELETE Object

API Description

You can use this API to delete an object from a specified bucket.

Method Definition

```
ObsClient.deleteObject
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
DeleteMarker	String	Whether the deleted object is a delete marker
VersionId	String	Object version ID

Sample Code

```
obsClient.deleteObject({
    Bucket:'bucketname',
    Key:'objectkey'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('VersionId-->' + result.InterfaceResult.VersionId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.6 DELETE Objects

API Description

You can use this API to delete objects from a specified bucket in a batch.

Method Definition

```
ObsClient.deleteObjects
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
Objects		Array	Mandatory	List of objects to be deleted
	Key	String	Mandatory	Object name

Field		Type	Optional or Mandatory	Description
	VersionId	String	Optional	Object version ID
Quiet		Boolean	Optional	Response mode of a batch deletion request. If this field is set to **false**, objects involved in the deletion will be returned. If this field is set to **true**, only objects failed to be deleted will be returned.

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
Deleteds		Array	List of successfully deleted objects
	Key	String	Name of the deleted object
	VersionId	String	Version ID of the deleted object
	DeleteMarker	String	Whether the deleted object is a delete marker
	DeleteMarkerVersionId	String	Version ID of the delete marker
Errors		Array	List of the objects that fail to be deleted
	Key	String	Name of the object that fails to be deleted
	VersionId	String	Version ID of the object that fails to be deleted
	Code	String	Error code of the deletion failure
	Message	String	Error message of the deletion failure

Sample Code

```
obsClient.deleteObjects({
    Bucket:'bucketname',
    Quiet:false,
    Objects:[{Key:'objectkey'},{Key:'objectkey2'}]
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
```



```
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Deleteds:');
            for(let i=0;i<result.InterfaceResult.Deleteds.length;i++){
                console.log('Deleted[' + i + ']:');
                console.log('Key-->'+result.InterfaceResult.Deleteds[i]['Key']);
                console.log('VersionId-->' + result.InterfaceResult.Deleteds[i]['VersionId']);
                console.log('DeleteMarker-->' + result.InterfaceResult.Deleteds[i]['DeleteMarker']);
                console.log('DeleteMarkerVersionId-->' + result.InterfaceResult.Deleteds[i]
['DeleteMarkerVersionId']);
            }
            console.log('Errors:');
            for(let i=0;i<result.InterfaceResult.Errors.length;i++){
                console.log('Error[' + i + ']:');
                console.log('Key-->' + result.InterfaceResult.Errors[i]['Key']);
                console.log('VersionId-->' + result.InterfaceResult.Errors[i]['VersionId']);
                console.log('Code-->' + result.InterfaceResult.Errors[i]['Code']);
                console.log('Message-->' + result.InterfaceResult.Errors[i]['Message']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.7 Obtain Object Metadata

API Description

You can use this API to send a HEAD request to the object of a specified bucket to obtain its metadata.

Method Definition

```
ObsClient.getObjectMetadata
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID
Origin	String	Optional	Origin of the cross-domain request specified by the pre-request. Generally, it is a domain name.

Field	Type	Optional or Mandatory	Description
RequestHeader	String	Optional	HTTP headers in the cross-domain request
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: • AES256
SseCKey	Buffer	Optional	Key used in SSE-C decryption, which is calculated by using AES256.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
LastModified	String	Time when the last modification was made to the object
ContentLength	String	Object size in bytes
ContentType	String	MIME type of the object
ETag	String	Object ETag
VersionId	String	Object version ID
WebsiteRedirectLocation	String	Location where the object is redirected to, when the bucket is configured with website hosting
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is **null**.
Restore	String	Restore status of the object in the OBS Archive storage class
AllowOrigin	String	If **Origin** in the request meets the CORS rules of the bucket, **AllowedOrigin** in the CORS rules is returned.
AllowHeader	String	If **AccessControlRequestHeaders** in the request meets the CORS rules of the bucket, **AllowedHeader** in the CORS rules is returned.

Field	Type	Description
AllowMethod	String	**AllowedMethod** in the CORS rules of the bucket
ExposeHeader	String	**ExposeHeader** in the CORS rules of the bucket
MaxAgeSeconds	String	**MaxAgeSeconds** in the CORS rules of the bucket
SseKms	String	Algorithm used in SSE-KMS decryption
SseKmsKey	String	Master key used in SSE-KMS decryption
SseC	String	Algorithm used in SSE-C decryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C decryption
Expiration	String	Expiration details
Metadata	Object	Customized metadata of the object
ObjectType	String	Whether the object is an appendable object
NextPosition	String	Start position for next appending

Sample Code

```
obsClient.getObjectMetadata({
    Bucket:'bucketname',
    Key : 'objectkey'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('ETag-->' + result.InterfaceResult.ETag);
            console.log('VersionId-->' + result.InterfaceResult.VersionId);
            console.log('ContentLength-->' + result.InterfaceResult.ContentLength);
            console.log('Expiration-->' + result.InterfaceResult.Expiration);
            console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
            console.log('LastModified-->' + result.InterfaceResult.LastModified);
            console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
        }else{
            console.log('Status-->' + result.CommonMsg.Status);
        }
    }
});
```

5.8 Modifying Object Metadata

API Description

You can use this API to send a PUT request to the object in a specified bucket to modify its metadata.

Method Definition

```
ObsClient.setObjectMetadata
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID
Origin	String	Optional	Origin of the cross-domain request specified by the pre-request. Generally, it is a domain name.
RequestHeader	String	Optional	HTTP headers in the cross-domain request
MetadataDirective	String	Mandatory	Metadata operation indicator. The value can be **REPLACE_NEW** or **REPLACE**. **REPLACE_NEW**: If the metadata of an existing value is replaced, a value is assigned to the metadata that does not have a value. The metadata that is not specified remains unchanged. **REPLACE**: Use the header field carried in the current request to replace the original metadata. The metadata that is not specified (except **StorageClass**) will be deleted.
CacheControl	String	Optional	Specifies the cache behavior of the web page when the object is downloaded.
ContentDisposition	String	Optional	Specifies the name of the object when it is downloaded.
ContentLanguage	String	Optional	Specifies the content language format when an object is downloaded.

Field	Type	Optional or Mandatory	Description
ContentEncoding	String	Optional	Specifies the content encoding format when an object is being uploaded.
ContentType	String	Optional	Object file type
Expires	String	Optional	Request expiration time
Metadata	Object	Optional	Customized metadata of the object
StorageClass	String	Optional	Specifies the storage class of an object. OBS provides three storage classes: Standard (STANDARD), Infrequent Access (WARM), and Archive (COLD). Therefore, values for this element can be **STANDARD**, **WARM**, and **COLD**. The spell of these values is case sensitive.
WebsiteRedirect-Location	String	Optional	When the bucket is configured with the website redirection, the request for obtaining the object can be redirected to another object or an external URL in the bucket.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
MetadataDirective	String	Metadata operation indicator. The value can be **REPLACE_NEW** or **REPLACE**.
CacheControl	String	Specifies the cache behavior of the web page when the object is downloaded.
Content-Disposition	String	Object ETag
Content-Encoding	String	Object version ID

Field	Type	Description
WebsiteRedirectLocation	String	Location where the object is redirected to, when the bucket is configured with website hosting
StorageClass	String	Storage class of the object. When the storage class is Standard, the value is **null**.
Content-Language	String	If **Origin** in the request meets the CORS rules of the bucket, **AllowedOrigin** in the CORS rules is returned.
Expires	String	If **AccessControlRequestHeaders** in the request meets the CORS rules of the bucket, **AllowedHeader** in the CORS rules is returned.
Metadata	Object	Customized metadata of the object

Sample Code

```
const metadata = { "test": "newmeta" };
const bucketName = 'bucketname';
const objectname = 'objectkey';

obsClient.setObjectMetadata({
      Bucket: bucketName,
      Key: objectname,
      ContentType: 'application/zip',
      StorageClass: obsClient.enums.StorageClassStandard,
      MetadataDirective: 'REPLACE_NEW',
      Metadata: metadata
},(err, result) => {
      if(err){
            console.error('Error-->' + err);
      }else{
            if(result.CommonMsg.Status < 300){
                  console.log('RequestId-->' + result.InterfaceResult.RequestId);
                  console.log('MetadataDirective-->' + result.InterfaceResult.MetadataDirective);
                  console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
                  console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
            }else{
                  console.log('Status-->' + result.CommonMsg.Status);
            }
      }
});
```

5.9 PUT Object acl

API Description

You can use this API to set the ACL for an object in a specified bucket.

Method Definition

```
ObsClient.setObjectAcl
```


Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			String	Mandatory	Bucket name
Key			String	Mandatory	Object name
VersionId			String	Optional	Object version ID
ACL			String	Optional	**Pre-defined ACL**
Owner			Object	Optional	Object owner
	ID		String	Mandatory	ID of the domain to which the object owner belongs
Delivered			Boolean	Optional	Whether the bucket ACL is applied to objects in the bucket
Grants			Array	Optional	List of grantees' permission information
	Grantee		Object	Mandatory	Grantee
		Type	String	Mandatory	**Available Grantee Types**
		ID	String	If **Type** is **CanonicalUser**, this field is optional. If **Type** is **Group**, this field must be **null**.	ID of the domain to which the grantee belongs

Field			Type	Optional or Mandatory	Description
		URI	String	If **Type** is **Group**, this field is mandatory. If **Type** is **CanonicalUser**, this field must be **null**.	**Available Grantee Groups**
	Permission		String	Mandatory	**Granted permission**

NOTE

- **Owner** and **Grants** must be used together. These two fields are mutually exclusive with **ACL**. When **ACL** is configured, these two fields are unavailable, and vice versa.
- You must set either the two fields or **ACL**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.setObjectAcl({
      Bucket:'bucketname',
      Key : 'objectkey',
      // Set the object owner.
      Owner:{ID:'ownerid'},
      Grants:[
          // Grant full control access to a specified user.
          { Grantee: { Type: 'CanonicalUser',ID: 'userid' }, Permission: obsClient.enums.PermissionFullControl },
          // Grant the READ permission to all users.
          { Grantee: { Type: 'Group', URI: obsClient.enums.GroupAllUsers }, Permission:
obsClient.enums.PermissionRead },
      ]
},(err, result) => {
      if(err){
            console.error('Error-->' + err);
      }else{
            if(result.CommonMsg.Status < 300){
```



```
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

NOTE

- Use the **Owner** parameter to specify the object owner and use the **Grants** parameter to grant permissions for authorized users.
- The owner or grantee ID needed in the ACL indicates the account ID, which can be viewed on the **My Credentials** page of OBS Console.
- OBS buckets support the following grantee group:
  - All users: ObsClient.enums.GroupAllUsers

5.10 GET Object acl

API Description

You can use this API to obtain the ACL of an object in a specified bucket.

Method Definition

```
ObsClient.getObjectAcl
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
VersionId		String	Object version ID
Owner		Object	Object owner
	ID	String	ID of the domain to which the object owner belongs

Field			Type	Description
Delivered			String	Whether the bucket ACL is applied to objects in the bucket
Grants			Array	List of grantees' permission information
	Grantee		Object	Grantee
		ID	String	ID of the domain to which the grantee belongs. This field is null when **Type** of **Grantee** is **Group**.
		URI	String	Grantee group. This field is null when **Type** of **Grantee** is **CanonicalUser**.
	Permission		String	Granted permission

Sample Code

```
obsClient.getObjectAcl({
    Bucket:'bucketname',
    Key : 'objectkey'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Owner[ID]-->' + result.InterfaceResult.Owner.ID);
            for(let i=0;i<result.InterfaceResult.Grants.length;i++){
                console.log('Grant[' + i + ']:');
                console.log('Grantee[ID]-->' + result.InterfaceResult.Grants[i]['Grantee']['ID']);
                console.log('Grantee[URI]-->' + result.InterfaceResult.Grants[i]['Grantee']['URI']);
                console.log('Permission-->' + result.InterfaceResult.Grants[i]['Permission']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.11 Initiate Multipart Upload

API Description

You can use this API to initialize a multipart upload in a specified bucket.

Method Definition

```
ObsClient.initiateMultipartUpload
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
ACL	String	Optional	**Pre-defined access control policy**
StorageClass	String	Optional	**Storage class** of the object
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
Expires	Number	Optional	Expiration time of the object, in days
ContentType	String	Optional	MIME type of the object
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: • kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be **null**.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	Buffer	Optional	Key used to encrypt the object in SSE-C mode, which is calculated by using AES256

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Bucket	String	Name of the bucket involved in the multipart upload

Field	Type	Description
Key	String	Name of the object to be uploaded
UploadId	String	Multipart upload ID
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.initiateMultipartUpload({
    Bucket:'bucketname',
    Key : 'objectkey',
    ContentType : 'text/plain'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Bucket-->' + result.InterfaceResult.Bucket);
            console.log('Key-->' + result.InterfaceResult.Key);
            console.log('UploadId-->' + result.InterfaceResult.UploadId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.12 PUT Part

API Description

After the multipart upload is initialized, you can use this API to upload a part to a specified bucket by using the multipart upload ID. Except for the part last uploaded whose size ranges from 0 to 5 GB, sizes of the other parts range from 100 KB to 5 GB. The upload part ID ranges from 1 to 10000.

Method Definition

```
ObsClient.uploadPart
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
PartNumber	Number	Mandatory	Part number, which ranges from 1 to 10000
UploadId	String	Mandatory	Multipart upload ID
ContentMD5	String	Optional	Base64-encoded MD5 value of the part to be uploaded. It is provided for the OBS server to verify data integrity.
Body	String or stream.Readable	Optional	Content of the part. Both character strings and instances of **stream.Readable** are supported.
SourceFile	String	Optional	Path to the source file of the part
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. This field is effective when **SourceFile** is set and its default value is **0**.
PartSize	Number	Optional	Size (in bytes) of a part in the source file. This field is effective when **SourceFile** is set and its default value is file size minus **Offset**. Except for the part lastly being uploaded whose size ranging from 0 to 5 GB, sizes of the other parts range from 100 KB to 5 GB.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	Buffer	Optional	Key used to encrypt the object in SSE-C mode, which is calculated by using AES256

NOTE

- **Body** and **SourceFile** cannot be used together.
- If both **Body** and **SourceFile** are **null**, the size of the uploaded object is **0** bytes.
- **Offset**, **PartSize**, and **SourceFile** can be used together to specify a part data in the to-be-uploaded source file.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag of current upload part
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.uploadPart({
    Bucket:'bucketname',
    Key : 'objectkey',
    UploadId : 'uploadid',
    PartNumber : 1,
    Body : 'Hello OBS'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('ETag-->' + result.InterfaceResult.ETag);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.13 PUT Part - Copy

API Description

After the multipart upload is initialized, you can use this API to copy a part to a specified bucket by using the multipart upload ID.

Method Definition

```
ObsClient.copyPart
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
PartNumber	Number	Mandatory	Part number, which ranges from 1 to 10000
UploadId	String	Mandatory	Multipart upload ID
CopySource	String	Mandatory	Parameter used to specify the source bucket, source object, and source object version ID which can be null. It is in the format of *SourceBucketName*/*SourceObjectName*?*versionId*=*SourceObjectVersionId*.
CopySourceRange	String	Optional	Copy source range. The value range is [0, source object length-1] and is in the format of *bytes=x-y*. If the maximum length of **CopySourceRange** is larger than the length of the source object minus 1, the length of the source object minus 1 is used.
SseC	String	Optional	Algorithm used to encrypt a target part in SSE-C mode. The value can be: • AES256
SseCKey	Buffer	Optional	Key used to encrypt a target part in SSE-C mode, which is calculated by using the AES256 algorithm.
CopySourceSseC	String	Optional	Algorithm used to decrypt a source object in SSE-C mode. The value can be: • AES256
CopySourceSseCKey	Buffer	Optional	Key used to decrypt a source object in SSE-C mode, which is calculated by using the AES256 algorithm.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag of the target part
LastModified	String	Time when the last modification was made to the target part
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.copyPart({
    Bucket:'bucketname',
    Key : 'objectkey',
    PartNumber : 1,
    UploadId : 'uploadid',
    CopySource : 'sourcebucketname/sourceobjectkey',
    CopySourceRange : 'bytes=0-10'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('LastModified-->' + result.InterfaceResult.LastModified);
            console.log('ETag-->' + result.InterfaceResult.ETag);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.14 List Parts

API Description

You can use this API to list the uploaded parts in a specified bucket by using the multipart upload ID.

Method Definition

```
ObsClient.listParts(parameter, callback)
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
UploadId	String	Mandatory	Multipart upload ID
MaxParts	Number	Optional	Maximum number of uploaded parts that can be listed per page
PartNumberMarker	Number	Optional	Part number after which listing uploaded parts begins. Only parts whose part numbers are larger than this value will be listed.

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
Bucket		String	Bucket name
Key		String	Object name
UploadId		String	Multipart upload ID
PartNumberMarker		String	Part number after which the listing uploaded parts begins, which is consistent with that set in the request
NextPartNumberMarker		String	Part number to start with upon the next request for listing uploaded parts
MaxParts		String	Maximum number of listed parts, which is consistent with that set in the request
IsTruncated		String	Whether all multipart uploads are returned for a request. If the field value is **true**, not all multipart uploads are returned. If the field value is **false**, all multipart uploads are returned.
Parts		Array	List of uploaded parts
	PartNumber	String	Part number

Field		Type	Description
	LastModified	String	Time when the part was last modified
	ETag	String	Part ETag
	Size	String	Part size
Initiator		Object	Initiator of the multipart upload
	ID	String	ID of the domain to which the owner belongs
Owner		Object	Owner of the multipart upload. It is the same as the initiator.
	ID	String	ID of the domain to which the owner belongs
StorageClass		String	Storage class of the object to be uploaded

Sample Code

```
obsClient.listParts({
    Bucket:'bucketname',
    Key : 'objectkey',
    UploadId : 'uploadid',
    MaxParts : 10
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Bucket-->' + result.InterfaceResult.Bucket);
            console.log('Key-->' + result.InterfaceResult.Key);
            console.log('UploadId-->' + result.InterfaceResult.UploadId);
            console.log('PartNumberMarker-->' + result.InterfaceResult.PartNumberMarker);
            console.log('NextPartNumberMarker-->' + result.InterfaceResult.NextPartNumberMarker);
            console.log('MaxParts-->' + result.InterfaceResult.MaxParts);
            console.log('IsTruncated-->' + result.InterfaceResult.IsTruncated);
            console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
            console.log('Initiator[ID]-->' + result.InterfaceResult.Initiator['ID']);
            console.log('Owner[ID]-->' + result.InterfaceResult.Owner['ID']);
            for(let i=0;i<result.InterfaceResult.Parts.length;i++){
                console.log('Part['+i+']:');
                console.log('PartNumber-->' + result.InterfaceResult.Parts[i]['PartNumber']);
                console.log('LastModified-->' + result.InterfaceResult.Parts[i]['LastModified']);
                console.log('ETag-->' + result.InterfaceResult.Parts[i]['ETag']);
                console.log('Size-->' + result.InterfaceResult.Parts[i]['Size']);
            }
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


5.15 Complete Multipart Upload

API Description

You can use this API to combine the uploaded parts in a specified bucket by using the multipart upload ID.

Method Definition

```
ObsClient.completeMultipartUpload
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
Key		String	Mandatory	Object name
UploadId		String	Mandatory	Multipart upload ID
Parts		Array	Mandatory	List of parts to be combined
	PartNumber	String	Mandatory	Part number
	ETag	String	Mandatory	Part ETag

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag calculated based on the ETags of all combined parts
Bucket	String	Bucket in which parts are combined
Key	String	Object name obtained after part combination
Location	String	URL of the object obtained after part combination

Field	Type	Description
VersionId	String	Version ID of the object obtained after part combination
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Master key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

```
obsClient.completeMultipartUpload({
    Bucket:'bucketname',
    Key : 'objectkey',
    UploadId : 'uploadid',
    Parts : [{PartNumber : 1, ETag : 'etag1'}, {PartNumber : 2, ETag : 'etag2'}]
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('Bucket-->' + result.InterfaceResult.Bucket);
            console.log('Key-->' + result.InterfaceResult.Key);
            console.log('Location-->' + result.InterfaceResult.Location);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.16 Abort Multipart Upload

API Description

You can use this API to abort a multipart upload in a specified bucket by using the multipart upload ID.

Method Definition

```
ObsClient.abortMultipartUpload
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
UploadId	String	Mandatory	Multipart upload ID

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
obsClient.abortMultipartUpload({
    Bucket:'bucketname',
    Key : 'objectkey',
    UploadId : 'uploadid'
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```

5.17 Restore an Archive Object

API Description

You can use this API to restore an object in the OBS Archive storage class in a specified bucket.

Method Definition

```
ObsClient.restoreObject
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Version ID of the to-be-restored object in the OBS Archive storage class
Days	Number	Mandatory	Retention period of the restored object, in days. The value ranges from 1 to 30.
Tier	String	Optional	**Restore option**

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
RestoreStatus	String	Restore status of the object. **AVAILABLE** indicates that the object can be downloaded. **INPROGRESS** indicates that the object is being restored.

Sample Code

```
obsClient.restoreObject({
    Bucket:'bucketname',
    Key : 'objectkey',
    Days : 1,
    Tier : obsClient.enums.RestoreTierExpedited
},(err, result) => {
    if(err){
        console.error('Error-->' + err);
    }else{
        if(result.CommonMsg.Status < 300){
            console.log('RequestId-->' + result.InterfaceResult.RequestId);
            console.log('RestoreStatus-->' + result.InterfaceResult.RestoreStatus);
        }else{
            console.log('Code-->' + result.CommonMsg.Code);
            console.log('Message-->' + result.CommonMsg.Message);
        }
    }
});
```


6 Other APIs

6.1 Creating a Signed URL

API Description

You can use this API to generate a URL whose **Query** parameter is carried with authentication information, by specifying the AK and SK, HTTP method, and request parameter. You can use a signed URL to perform specific operations on OBS.

Method Definition

```
ObsClient.createSignedUrlSync
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Method	String	Mandatory	HTTP method. Possible values are: • GET • POST • PUT • DELETE • HEAD
Bucket	String	Optional	Bucket name
Key	String	Optional	Object name

Field	Type	Optional or Mandatory	Description
SpecialParam	String	Optional	Special operator, which indicates the sub-resource to be operated. Possible values are: • versions • uploads • location • storageinfo • quota • storagePolicy • acl • append • logging • policy • lifecycle • website • versioning • cors • notification • tagging • delete • restore
Expires	Number	Optional	Expiration time of the signed URL, in seconds. The default value is **300**.
Headers	Object	Optional	Headers in the request
QueryParams	Object	Optional	Query parameters in the request

Returned Result

Field	Type	Description
SignedUrl	String	Signed URL
ActualSignedRequestHeaders	Object	Actual headers in the request initiated by using the signed URL

Sample Code

```
// Generate a signed URL for creating a bucket.
var createBucketResult = obsClient.createSignedUrlSync({Method : 'PUT', Bucket : 'bucketname'});
console.log('SignedUrl-->' + createBucketResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(createBucketResult['ActualSignedRequestHeaders']));

// Generate a signed URL for uploading an object.
var putObjectResult = obsClient.createSignedUrlSync({Method : 'PUT', Bucket : 'bucketname', Key :
'objectkey', Headers : {'Content-Type' : 'text/plain'}});
console.log('SignedUrl-->' + putObjectResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(putObjectResult['ActualSignedRequestHeaders']));

// Generate a signed URL for setting an object ACL.
var setObjectAclResult = obsClient.createSignedUrlSync({Method : 'PUT', Bucket : 'bucketname', Key :
'objectkey', SpecialParam: 'acl', Headers: {'x-obs-acl' : 'public-read'}});
console.log('SignedUrl-->' + setObjectAclResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(setObjectAclResult['ActualSignedRequestHeaders']));

// Generate a signed URL for downloading an object.
var getObjectResult = obsClient.createSignedUrlSync({Method : 'GET', Bucket : 'bucketname', Key :
'objectkey'});
console.log('SignedUrl-->' + getObjectResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(getObjectResult['ActualSignedRequestHeaders']));

// Generate a signed URL for deleting an object.
var deleteObjectResult = obsClient.createSignedUrlSync({Method : 'DELETE', Bucket : 'bucketname', Key :
'objectkey'});
console.log('SignedUrl-->' + deleteObjectResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(deleteObjectResult['ActualSignedRequestHeaders']));

// Generate a signed URL for deleting a bucket.
var deleteBucketResult = obsClient.createSignedUrlSync({Method : 'DELETE', Bucket : 'bucketname'});
console.log('SignedUrl-->' + deleteBucketResult['SignedUrl']);
console.log('ActualSignedRequestHeaders-->' +
JSON.stringify(deleteBucketResult['ActualSignedRequestHeaders']));
```

6.2 Generating Browser-Based Upload Parameters with Authentication Information

API Description

You can use this API to generate parameters for authentication. The parameters can be used to upload data through POST operations based on a browser.

NOTE

There are two request parameters generated:

- **Policy**, which corresponds to the **policy** field in the form
- **Signature**: which corresponds to the **signature** field in the form

Method Definition

```
ObsClient.createPostSignatureSync
```


Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Optional	Bucket name
Key	String	Optional	Object name, corresponds to the **key** field in the form
Expires	Number	Optional	Expiration time of the browser-based authentication (unit: seconds). The default value is **300**.
FormParams	Object	Optional	Other parameters of the browser-based upload except for **key**, **policy**, and **signature**. Possible values are: • acl • cache-control • content-type • content-disposition • content-encoding • expires

Returned Result

Field	Type	Description
OriginPolicy	String	**policy** not encoded by Base64. This parameter can only be used for verification.
Policy	String	**policy** in a form
Signature	String	**signature** in the form

Sample Code

```
var formParams = {acl: 'public-read', 'content-type': 'text/plain'};
var res = obsClient.createPostSignatureSync({Bucket: 'bucketname', Key: 'objectkey', Expires:3600,
FormParams: formParams});

console.log('Policy-->' + res['Policy']);
console.log('Signature-->' + res['Signature']);
```


6.3 Performing a Resumable Upload

API Description

This API is an encapsulated and enhanced version of multipart upload, and aims to eliminate large file upload failures caused by poor network conditions and program breakdowns.

Method Definition

```
ObsClient.uploadFile
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
UploadFile	String	Mandatory	Local file to be uploaded
PartSize	Number	Optional	Part size, in bytes. The value ranges from 100 KB to 5 GB and is 5 MB by default.
TaskNum	Number	Optional	Maximum number of threads that can be concurrently executed for upload. The default value is **20**.
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress **NOTE** This callback function contains the following parameters in sequence: number of uploaded bytes, total bytes, and used time (unit: second).
ResumeCallback	Function	Optional	Callback function used to obtain the control parameter for canceling a resumable upload **NOTE** • This callback function contains a control parameter used for canceling resumable uploads. • By calling the cancel method of this control parameter, you can pause a resumable upload.

Field	Type	Optional or Mandatory	Description
EnableCheckpoint	Boolean	Optional	Whether to enable the resumable upload mode. The default value is **false**, which indicates that this mode is disabled.
CheckpointFile	String	Optional	File used to record the upload progress. This parameter is effective only in the resumable upload mode. If this parameter is **null**, the file will be in the same directory as the local file to be uploaded.
EnableCheckSum	Boolean	Optional	Whether to verify the content of the to-be-uploaded file. This parameter is effective only in the resumable mode. The default value is **false**, which indicates that the content will not be verified.
ContentType	String	Optional	MIME type of the object
ACL	String	Optional	**Pre-defined access control policy**
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: • kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be **null**.
sseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	String	Optional	Key used to encrypt the object in SSE-C mode, which is calculated by using AES256
Metadata	Object	Optional	Customized metadata of the object

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag calculated based on the ETags of all combined parts
Bucket	String	Bucket in which parts are combined
Key	String	Object name obtained after part combination
Location	String	URL of the object obtained after part combination
VersionId	String	Version ID of the object obtained after part combination
SseKms	String	Algorithm used in SSE-KMS encryption
SseKmsKey	String	Master key used in SSE-KMS encryption
SseC	String	Algorithm used in SSE-C encryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C encryption

Sample Code

This example uploads the **example** file to the **examplebucket** bucket in a resumable upload and prints the progress.

```
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');

// Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
// Obtain an AK/SK pair on the management console by referring to https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
const AK = process.env.AccessKeyID
const SK = process.env.SecretAccessKey
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, obtain them using environment variables.
const security_token= process.env.SecurityToken
// Set server to the endpoint corresponding to the bucket. EU-Dublin is used here as an example. Replace it with the one in your actual situation.
const server = "https://obs.eu-west-101.myhuaweicloud.com"

var obsClient = new ObsClient({
    access_key_id: AK,
    secret_access_key: SK,
    server: server,
});

const Bucket = 'examplebucket'
// Name of the object after being uploaded
const Key = 'objectname'
// Local file to be uploaded
const UploadFile = 'D:\\example'
```



```
// Part size, in bytes. The value ranges from 100 KB to 5 GB and is 5 MB by default.
const PartSize = 10 * 1024 * 1024
// Maximum number of concurrent part uploads. The default value is 20.
const TaskNum = 10
// Define the control parameter for canceling the resumable upload.
var hook;
// Perform the resumable upload.
try {
    obsClient.uploadFile({
        Bucket,
        Key,
        UploadFile,
        PartSize,
        TaskNum,
        // Enable the resumable upload.
        EnableCheckpoint: true,
        // Progress callback
        ProgressCallback: function (transferredAmount, totalAmount, totalSeconds) {
            // Print the upload speed in KB/s.
            console.log(transferredAmount * 1.0 / totalSeconds / 1024);
            // Print the upload percentage.
            console.log(transferredAmount * 100.0 / totalAmount);
            // Pause the upload when the progress reaches 50%.
            // if(hook && (transferredAmount / totalAmount) > 0.5){
            //     // Pause the resumable upload.
            //     hook.cancel();
            // }
        },
        ResumeCallback: function (resumeHook) {
            // Obtain the control parameter for canceling the resumable upload.
            hook = resumeHook;
        }
    }, (err, result) => {
        if (err) {
            console.log('UploadFile Failed')
            console.error('Error-->' + err);
        } else {
            // If status code 2xx is returned, the API is called successfully. Otherwise, the API call fails.
            if (result.CommonMsg.Status < 300) {
                console.log('UploadFile Succeeded')
                console.log('RequestId-->' + result.InterfaceResult.RequestId);
                console.log('Bucket-->' + result.InterfaceResult.Bucket);
                console.log('Key-->' + result.InterfaceResult.Key);
                console.log('Location-->' + result.InterfaceResult.Location);
            } else {
                console.log('UploadFile Failed')
                console.log('ErrorCode-->' + result.CommonMsg.Code);
                console.log('ErrorMessage-->' + result.CommonMsg.Message);
            }
        }
    });
} catch (error) {
    console.log('UploadFile Failed')
    console.error('Error-->' + error);
}
```

6.4 Performing a Resumable Download

API Description

This API is an encapsulated and enhanced version of multipart download, and aims to eliminate large file download failures caused by poor network conditions and program breakdowns.

Method Definition

```
ObsClient.downloadFile
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
DownloadFile	String	Optional	Local path to which the object is downloaded. If this parameter is **null**, the downloaded object is saved in the directory where the program is executed.
PartSize	Number	Optional	Part size, in bytes. The value ranges from 100 KB to 5 GB and is 5 MB by default.
TaskNum	Number	Optional	Maximum number of threads that can be concurrently executed for download. The default value is **20**.
ProgressCallback	Function	Optional	Callback function for obtaining the download progress **NOTE** This callback function contains the following parameters in sequence: number of downloaded bytes, total bytes, and used time (unit: second).
ResumeCallback	Function	Optional	Callback function used to obtain the control parameter for canceling a resumable download **NOTE** • This callback function contains a control parameter used for canceling resumable downloads. • By calling the cancel method of this control parameter, you can pause a resumable download.
EnableCheckpoint	Boolean	Optional	Whether to enable the resumable upload mode. The default value is **false**, which indicates that this mode is disabled.

Field	Type	Optional or Mandatory	Description
CheckpointFile	String	Optional	File used to record the download progress. This parameter is effective only in the resumable download mode. If this parameter is **null**, the file will be in the same directory as the local directory of the downloaded file.
VersionId	String	Optional	Object version ID
IfModifiedSince	String	Optional	Returns the object if it is modified after the time specified by this parameter; otherwise, an error code is returned.
IfUnmodifiedSince	String	Optional	Returns the object if it remains unchanged since the time specified by this parameter; otherwise, an error code is returned.
IfMatch	String	Optional	Returns the source object if its ETag is the same as the one specified by this parameter; otherwise, an error code is returned.
IfNoneMatch	String	Optional	Returns the source object if its ETag is different from the one specified by this parameter; otherwise, an error code is returned.
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: • AES256
SseCKey	String	Optional	Key used in SSE-C decryption, which is calculated by using AES256

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server

Field	Type	Description
DeleteMarker	String	Whether the deleted object is a delete marker
LastModified	String	Time when the last modification was made to the object
CacheControl	String	**Cache-Control** header in the response
ContentDisposition	String	**Content-Disposition** header in the response
ContentEncoding	String	**Content-Encoding** header in the response
ContentLanguage	String	**Content-Language** header in the response
ContentType	String	MIME type of the object
Expires	String	**Expires** header in the response
ETag	String	Object ETag
VersionId	String	Object version ID
WebsiteRedirectLocation	String	Location where the object is redirected to, when the bucket is configured with website hosting
StorageClass	String	Object storage class. If the storage class is Standard, this parameter is left blank.
Restore	String	Restore status of the object in the OBS Archive storage class
SseKms	String	Algorithm used in SSE-KMS decryption
SseKmsKey	String	Master key used in SSE-KMS decryption
SseC	String	Algorithm used in SSE-C decryption
SseCKeyMd5	String	MD5 value of the key used in SSE-C decryption
Expiration	String	Expiration details
Metadata	Object	Customized metadata of the object

Sample Code

This example downloads **objectname** from **examplebucket** in a resumable download and prints the progress.

```
// Use npm to install the client.
var ObsClient = require('esdk-obs-nodejs');
```



```
// Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
// Obtain an AK/SK pair on the management console by referring to https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
const AK = process.env.AccessKeyID
const SK = process.env.SecretAccessKey
// (Optional) If you use a temporary AK/SK pair and a security token to access OBS, obtain them using environment variables.
const security_token= process.env.SecurityToken
// Set server to the endpoint corresponding to the bucket. EU-Dublin is used here as an example. Replace it with the one in your actual situation.
const server = "https://obs.eu-west-101.myhuaweicloud.com"

var obsClient = new ObsClient({
    access_key_id: AK,
    secret_access_key: SK,
    server: server,
});
const Bucket = 'examplebucket'
// Name of the object to be downloaded
const Key = 'objectname'
// Set the local path the object will be downloaded to.
const DownloadFile = 'D:\\example'
// Set the part size to 10 MB.
const PartSize = 10 * 1024 * 1024
// Number of concurrent downloads
const TaskNum = 10
// Define the control parameter used for canceling the resumable download.
var hook;
// Perform the resumable download.
try {
    obsClient.downloadFile({
        Bucket,
        Key,
        DownloadFile,
        PartSize,
        TaskNum,
        // Enable the resumable download.
        EnableCheckpoint: true,
        // Progress callback
        ProgressCallback: function (transferredAmount, totalAmount, totalSeconds) {
            // Print the download speed in KB/s.
            console.log(transferredAmount * 1.0 / totalSeconds / 1024);
            // Print the download percentage.
            console.log(transferredAmount * 100.0 / totalAmount);
            // Pause the download when the progress reaches 50%.
            // if(hook && (transferredAmount / totalAmount) > 0.5){
            //     // Pause the resumable download.
            //     hook.cancel();
            // }
        },
        ResumeCallback : function(resumeHook){
            // Obtain the control parameter used for canceling the resumable download.
            hook = resumeHook;
        }
    }, (err, result) => {
        if (err) {
            console.log('DownloadFile Failed')
            console.error('Error-->' + err);
        } else {
            if (result.CommonMsg.Status < 300) {
                console.log('DownloadFile Succeeded')
                console.log('RequestId-->' + result.InterfaceResult.RequestId);
                console.log('LastModified-->' + result.InterfaceResult.LastModified);
                console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
            } else {
                console.log('DownloadFile Failed')
                console.log('ErrorCode-->' + result.CommonMsg.Code);
                console.log('ErrorMessage-->' + result.CommonMsg.Message);
```



```
        }
      }
    });
} catch (error) {
    console.log('DownloadFile Failed')
    console.error('Error-->' + error);
}
```


A Change History

Release Date	What's New
2023-03-14	This is the first official release.

