

# Object Storage Service

## Node.js SDK API Reference

Issue 07  
Date 2019-03-30



**Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.**

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

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

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

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

---

# Contents

---

<b>1 Overview</b>	<b>1</b>
<b>2 Initialization</b>	<b>2</b>
2.1 Initializing an Instance of ObsClient	2
2.2 Initializing Logs	4
2.3 Request Objects	5
2.4 SDK Common Result Objects	5
2.5 Result Returned via a Callback Function	6
2.6 Result Returned via the Promise Object	7
<b>3 Pre-defined Constants</b>	<b>8</b>
3.1 Permission Types	8
3.2 Available Grantee Types	9
3.3 Available Grantee Groups	9
3.4 Pre-defined Access Control Policies	10
3.5 Storage Classes	10
3.6 Restore Options	11
3.7 Event Types	11
3.8 Metadata Replication Policy	12
<b>4 Bucket-Related APIs</b>	<b>13</b>
4.1 PUT Bucket	13
4.2 GET Buckets	15
4.3 HEAD Bucket	16
4.4 DELETE Bucket	17
4.5 GET Objects	18
4.6 GET Object versions	21
4.7 List Multipart uploads	24
4.8 Obtain Bucket Metadata	27
4.9 GET Bucket location	29
4.10 GET Bucket storageinfo	30
4.11 PUT Bucket quota	31
4.12 GET Bucket quota	32
4.13 Set Bucket storagePolicy	33
4.14 GET Bucket storagePolicy	34

4.15 PUT Bucket acl.....	35
4.16 GET Bucket acl.....	37
4.17 PUT Bucket logging.....	38
4.18 GET Bucket logging.....	40
4.19 PUT Bucket policy.....	42
4.20 GET Bucket policy.....	43
4.21 DELETE Bucket policy.....	44
4.22 PUT Bucket lifecycle.....	44
4.23 GET Bucket lifecycle.....	49
4.24 DELETE Bucket lifecycle.....	51
4.25 PUT Bucket website.....	52
4.26 GET Bucket website.....	54
4.27 DELETE Bucket website.....	56
4.28 PUT Bucket versioning.....	57
4.29 GET Bucket versioning.....	58
4.30 PUT Bucket cors.....	59
4.31 GET Bucket cors.....	61
4.32 DELETE Bucket cors.....	63
4.33 PUT Bucket notification.....	64
4.34 GET Bucket notification.....	66
4.35 PUT Bucket tagging.....	68
4.36 GET Bucket tagging.....	69
4.37 DELETE Bucket tagging.....	70
<b>5 Object-Related APIs.....</b>	<b>72</b>
5.1 PUT Object.....	72
5.2 Append Object.....	75
5.3 GET Object.....	77
5.4 PUT Object - Copy.....	82
5.5 DELETE Object.....	85
5.6 DELETE Objects.....	86
5.7 Obtain Object Metadata.....	88
5.8 Modifying Object Metadata.....	91
5.9 PUT Object acl.....	94
5.10 GET Object acl.....	96
5.11 Initiate Multipart Upload.....	97
5.12 PUT Part.....	99
5.13 PUT Part - Copy.....	101
5.14 List Parts.....	103
5.15 Complete Multipart Upload.....	106
5.16 Abort Multipart Upload.....	107
5.17 POST Object Restore.....	108
<b>6 Other APIs.....</b>	<b>110</b>

---

6.1 Creating a Signed URL.....	110
6.2 Generating Browser-Based Upload Parameters with Authentication Information.....	112
6.3 Performing a Resumable Upload.....	114
6.4 Performing a Resumable Download.....	116
<b>A Change History.....</b>	<b>120</b>

# 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 usage of OBS Node.js SDK (such as installation, initialization, development, and FAQs), application scenarios of interfaces, and code examples in various scenarios, see the [Object Storage Service Node.js SDK Developer Guide](#).

# 2 Initialization

## 2.1 Initializing an Instance of ObsClient

### 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, <code>https://your-endpoint:443</code> . You can click <a href="#">here</a> to view the endpoints enabled for OBS.
max_retry_count	Number	Optional	Maximum number of retries when an HTTP/HTTPS connection is abnormal. The default value is <b>3</b> .

Field	Type	Optional or Mandatory	Description
timeout	Number	Optional	Timeout period (in seconds) of an HTTP/HTTPS request. The default value is <b>60</b> .
ssl_verify	Boolean or String	Optional	Whether to verify the server certificate. Possible values are: <ul style="list-style-type: none"> <li>• Path to the server-side root certificate file in <b>.pem</b> format</li> <li>• <b>true</b>: The default CAs are used to verify the server-side certificate.</li> <li>• <b>false</b>: The server-side certificates will not be verified.</li> </ul> The default value is <b>false</b> .
long_conn_param	Number	Optional	Persistent connection mode (in seconds) If the value is equal to or larger than <b>0</b> , the persistent connection mode is enabled and this value is used as the <b>initial delay</b> 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 <b>false</b> .

## Sample Code

```
// Introduce 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 ObsClient instance.
var obsClient = new ObsClient({
  access_key_id: '*** Provide your Access Key ***', // Configure the AK.
  secret_access_key: '*** Provide your Secret Key ***', // Configure the SK.
  server: 'https://your-endpoint', // Configure the endpoint.
  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 <b>ObsClient</b> , 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 <b>debug</b> , <b>info</b> , <b>warn</b> , and <b>error</b> .
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 <b>300</b> , the operation succeeds. Otherwise, the operation fails.
	Code	String Error code returned by the OBS server. If <b>Status</b> is smaller than <b>300</b> , the value is <b>null</b> .
	Message	String Error description returned by the OBS server. If <b>Status</b> is smaller than <b>300</b> , the value is <b>null</b> .

Field		Type	Description
	HostId	String	Server ID. If <b>Status</b> is smaller than <b>300</b> , the value is <b>null</b> .
	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 <b>Status</b> is smaller than <b>300</b> , the value is <b>null</b> .
InterfaceResult		Object	Result returned after an operation is successfully performed. If <b>Status</b> is smaller than <b>300</b> , the value is <b>null</b> .
	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	<p>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.</p> <p>A grantee with this permission for an object can obtain the object content and metadata.</p>
obs.js	ObsClient.enums.PermissionWrite	String	<p>A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket.</p> <p>This permission is not applicable to objects.</p>
obs.js	ObsClient.enums.PermissionReadAcp	String	<p>A grantee with this permission can obtain the ACL of a bucket or object.</p> <p>A bucket or object owner has this permission permanently.</p>

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

## 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 Event Types

Module	Access Method	Type	Description
obs.js	ObsClient.enums.EventObjectCreatedAll	String	All events for creating objects
obs.js	ObsClient.enums.EventObjectCreatedPut	String	PUT Object events
obs.js	ObsClient.enums.EventObjectCreatedPost	String	POST Object events
obs.js	ObsClient.enums.EventObjectCreatedCopy	String	PUT Object - Copy events
obs.js	ObsClient.enums.EventObjectCreatedCompleteMultipartUpload	String	Complete Multipart Upload events
obs.js	ObsClient.enums.EventObjectRemovedAll	String	All events for deleting objects
obs.js	ObsClient.enums.EventObjectRemovedDelete	String	Events for deleting objects by specifying object version IDs
obs.js	ObsClient.enums.EventObjectRemovedDeleteMarkerCreated	String	Events for deleting objects without specifying object version IDs, when versioning is enabled

## 3.8 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	<p>Bucket name.</p> <p>A bucket name must comply with the following rules:</p> <ul style="list-style-type: none"> <li>• Contains 3 to 63 characters, starts with a digit or letter, and supports only lowercase letters, digits, hyphens (-), and periods (.)</li> <li>• Cannot be an IP-like address.</li> <li>• Cannot start or end with a hyphen (-) or period (.)</li> <li>• Cannot contain two consecutive periods (.), for example, <b>my..bucket</b>.</li> <li>• Cannot contain periods (.) and hyphens (-) adjacent to each other, for example, <b>my-.bucket</b> or <b>my.-bucket</b>.</li> </ul>
ACL	String	Optional	<b>Pre-defined access control policy</b> that can be specified during the bucket creation
StorageClass	String	Optional	Bucket <b>storage class</b> that can be specified during the bucket creation
Location	String	Mandatory unless the region where the OBS service resides is not the default region.	<p>Bucket location</p> <p>This parameter is not required if the endpoint belongs to the default North China 1 region (cn-north-1). This parameter is a must if the endpoint belongs to any other regions. Click <a href="#">here</a> to query currently valid regions. For details about OBS regions and endpoints, see <a href="#">Regions and Endpoints</a>.</p>

## 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.StorageClassCold
}, (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 <b>Delimiter</b> parameter, the character string from the first character to the first <b>Delimiter</b> parameter in the object name is grouped under a single result element, <b>CommonPrefix</b> . (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 <b>true</b> , not all objects are returned. If the field value is <b>false</b> , 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 <b>Delimiter</b> parameter (if specified)
	Prefix	String	Object name prefix grouped according to the <b>Delimiter</b> 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 <b>Delimiter</b> parameter, the character string from the first character to the first <b>Delimiter</b> parameter in the object name is grouped under a single result element, <b>CommonPrefix</b> . (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 <b>KeyMarker</b> .  If the value of <b>VersionIdMarker</b> is not a version ID specified by <b>KeyMarker</b> , <b>VersionIdMarker</b> 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 <b>true</b> , not all versioning objects are returned. If the field value is <b>false</b> , 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 <b>NextKeyMarker</b> 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 <b>true</b> , 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 <b>true</b> , 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 <b>Delimiter</b> parameter (if specified)	

Field	Type	Description
Prefix	String	Object name prefix grouped according to the <b>Delimiter</b> 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 <b>Delimiter</b> parameter, the character string from the first character to the first <b>Delimiter</b> parameter in the object name is grouped under a single result element, <b>CommonPrefix</b> . (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 <b>KeyMarker</b> , so that multipart uploads after <b>UploadIdMarker</b> of <b>KeyMarker</b> 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 <b>NextKeyMarker</b> 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 <b>true</b> , not all multipart uploads are returned. If the field value is <b>false</b> , 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 <b>Delimiter</b> parameter (if specified)
	Prefix	String	Object name prefix grouped according to the <b>Delimiter</b> 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 <b>Origin</b> in the request meets the CORS rule of the server, <b>AllowedOrigin</b> in the CORS configuration of the server is returned.
AllowHeader	String	If <b>RequestHeader</b> in the request meets the CORS rule of the server, <b>AllowedHeader</b> in the CORS configuration of the server is returned.
AllowMethod	String	<b>AllowedMethod</b> in the CORS rule of the server
ExposeHeader	String	<b>ExposeHeader</b> in the CORS rule of the server
MaxAgeSeconds	String	<b>MaxAgeSeconds</b> 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	<b>Storage class</b> 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
StorageClasses	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	<b>Pre-defined access control policy</b>
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	<b>Available Grantee Types</b>
	ID	String	If <b>Type</b> is <b>CanonicalUser</b> , this field is mandatory. If <b>Type</b> is <b>Group</b> , this field must be <b>null</b> .	ID of the domain to which the grantee belongs

Field		Type	Optional or Mandatory	Description
	URI	String	If <b>Type</b> is <b>Group</b> , this field is mandatory. If <b>Type</b> is <b>CanonicalUser</b> , this field must be <b>null</b> .	<b>Available Grantee Groups</b>
	Permission	String	Mandatory	<b>Granted permission</b>
	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	
	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 <b>Type</b> of <b>Grantee</b> is <b>Group</b> .
	ID	String	ID of the domain to which the grantee belongs. This field is null when <b>Type</b> of <b>Grantee</b> is <b>Group</b> .

Field		Type	Description
	URI	String	Grantee group. This field is null when <b>Type</b> of <b>Grantee</b> is <b>CanonicalUser</b> .
	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	<b>Grantee type</b>
			ID	String	If <b>Type</b> is <b>Canonical User</b> , this field is mandatory. If <b>Type</b> is <b>Group</b> , this field must be <b>null</b> .	ID of the domain to which the grantee belongs
			URI	String	If <b>Type</b> is <b>Group</b> , this field is mandatory. If <b>Type</b> is <b>Canonical User</b> , this field must be <b>null</b> .	<b>Grantee group</b>
	Permission	String	Optional	<b>Granted permission</b>		

## 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	Grantee	Object	Grantee
		ID	String	ID of the domain to which the grantee belongs. This field is null when <b>Type</b> of <b>Grantee</b> is <b>Group</b> .
		URI	String	Grantee group. This field is null when <b>Type</b> of <b>Grantee</b> is <b>CanonicalUser</b> .
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']);
        }
      }
    }
    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 about the format, see <a href="#">Policy Format</a> .

### 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	<b>Storage class</b> of the object after transition <b>NOTE</b> The Standard storage class is not supported.
		Date	String	This parameter is mandatory if <b>Days</b> is not set.	Date when an object will be transitioned The value must conform with the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01 T00:00:00Z.

Field			Type	Optional or Mandatory	Description
		Days	Number	This parameter is mandatory if <b>Date</b> 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 <b>Days</b> 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, 2018-01-01 T00:00:00Z.
		Days	Number	This parameter is mandatory if <b>Date</b> 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: <ul style="list-style-type: none"> <li>• Enabled</li> <li>• Disabled</li> </ul>
	NoncurrentVersionTransitions	Array	Optional	List of noncurrent object version transition policies

Field			Type	Optional or Mandatory	Description
		StorageClass	String	Mandatory	<b>Storage class</b> 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: <ul style="list-style-type: none"> <li>• <b>http</b> (default)</li> <li>• <b>https</b></li> </ul>
ErrorDocument		Object	Optional	Error page settings
	Key	String	Optional	Page to return to when a 4XX 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 <b>index.html</b> and you request for <b>samplebucket/images/</b> , the data that is returned will be the object with the key name <b>images/index.html</b> in the <b>samplebucket</b> 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: <ul style="list-style-type: none"> <li>• http</li> <li>• https</li> </ul>
	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 <b>ReplaceKeyPrefixWith</b> .
	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 <b>index.html</b> and you request for <b>samplebucket/images/</b> , the data that is returned will be the object with the key name <b>images/index.html</b> in the <b>samplebucket</b> 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 <b>ReplaceKeyPrefixWith</b> .
	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: <ul style="list-style-type: none"> <li>Enabled</li> <li>Suspended</li> </ul>

## 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: <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• HEAD</li> <li>• POST</li> <li>• DELETE</li> </ul>
	AllowedOrigin	Array of Strings	Mandatory	Origins (character strings representing domain names) allowed by the CORS rule. The value can contain one wildcard character (*). Each <b>AllowedOrigin</b> 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 <b>AllowedHeader</b> 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 notification

### API Description

You can use this API to set event notification for a bucket. The user will be notified of all specified operations performed on the bucket.

### Method Definition

`ObsClient.setBucketNotification`

### Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		String	Mandatory	Bucket name
TopicConfigurations		Array	Optional	Configuration list of the Simple Notification Service (SMS)
	ID	String	Optional	Configuration ID of the SMS
	Topic	String	Mandatory	URN of the event notification topic. After detecting a specific event in a bucket, OBS sends a message to the topic.
	Event	Array	Mandatory	<b>Type</b> of events that need to be notified
	Filter	Object	Optional	Filtering rule information
	FilterRules	Array	Optional	List of filtering rules
		Name	String	Optional
	Value	String	Optional	Keyword of object name for filtering objects by prefix or suffix

Field		Type	Optional or Mandatory	Description	
FunctionGraphConfigurations		Array	Optional	FunctionGraph configuration list	
	ID	String	Optional	FunctionGraph configuration ID	
	FunctionGraph	String	Mandatory	URN of FunctionGraph. When OBS detects a specific event in the bucket, it sends a message to FunctionGraph and calls FunctionGraph.	
	Event	Array	Mandatory	List of <b>event types</b> that need to send messages to FunctionGraph.	
	Filter	Object	Optional	Filtering rule information	
		FilterRules	Array	Optional	List of filtering rules
			Name	String	Optional
		Value	String	Optional	Keyword of object name for filtering objects by prefix or suffix

### Returned Result (InterfaceResult)

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

### Sample Code

```
obsClient.setBucketNotification({
  Bucket : 'bucketname',
  TopicConfigurations : [
    {
      ID : '001',
      Topic : 'your topic',
      Event : [obsClient.enums.EventObjectCreatedAll],
    }
  ]
})
```

```

        Filter : {
          FilterRules : [{Name : 'prefix', Value: 'smn'}, {Name : 'suffix', Value: '.jpg'}]
        }
      ]
    },(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 notification

### API Description

You can use this API to obtain the event notification settings of a bucket.

### Method Definition

ObsClient.getBucketNotification

### 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	
TopicConfigurations	Array	Configuration list of the Simple Notification Service (SMS)	
	ID	String	Configuration ID of the SMS
	Topic	String	URN of the event notification topic. After detecting a specific event in a bucket, OBS sends a message to the topic.

Field		Type	Description	
	Event	Array	Type of events that need to be notified.	
	Filter	Object	Filtering rule information	
	FilterRules	Array	List of filtering rules	
		Name	String	Prefix or suffix of object names for filtering
		Value	String	Keyword of object name for filtering objects by prefix or suffix
FunctionGraphConfigurations		Array	FunctionGraph configuration list	
	ID	String	FunctionGraph configuration ID	
	FunctionGraph	String	URN of FunctionGraph. When OBS detects a specific event in the bucket, it sends a message to FunctionGraph and calls FunctionGraph.	
	Event	Array	List of event types that need to send messages to FunctionGraph.	
	Filter	Object	Filtering rule information	
	FilterRules	Array	List of filtering rules	
		Name	String	Prefix or suffix of object names for filtering
		Value	String	Keyword of object name for filtering objects by prefix or suffix

## Sample Code

```

obsClient.getBucketNotification({
  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.TopicConfigurations){
        for(let j=0;j<result.InterfaceResult.TopicConfigurations;j++){
          console.log('ID-->' + result.InterfaceResult.TopicConfigurations[j].ID);
          console.log('Topic-->' + result.InterfaceResult.TopicConfigurations[j].Topic);
          console.log('Event-->' + result.InterfaceResult.TopicConfigurations[j].Event);
          if(result.InterfaceResult.TopicConfigurations[j].Filter){
            for(let i=0;i<result.InterfaceResult.TopicConfigurations[j].Filter.FilterRules.length;i++){
              console.log('FilterRule[' + i + '][Name]-->' +
result.InterfaceResult.TopicConfigurations[j].Filter.FilterRules[i].Name);
              console.log('FilterRule[' + i + '][Value]-->' +
result.InterfaceResult.TopicConfigurations[j].Filter.FilterRules[i].Value);
            }
          }
        }
      }
    }
  }
}

```



## 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);
    }
  }
});
```

## 4.36 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);
    }
  }
});
```

## 4.37 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	<b>Pre-defined access control policy</b> specified for the object
StorageClass	String	Optional	<b>Storage class</b> 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 <b>stream.Readable</b> are supported.
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. This field is effective when <b>SourceFile</b> is set and its default value is <b>0</b> .
SourceFile	String	Optional	Path to the source file of the object
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress <b>NOTE</b> 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
SuccessActionRedirect	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 <b>SourceFile</b> 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: <ul style="list-style-type: none"> <li>kms</li> </ul>
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be <b>null</b> .

Field	Type	Optional or Mandatory	Description
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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 <b>0</b> when you create an appendable object.
ACL	String	Optional	<b>Pre-defined access control policy</b> specified during the object creation
StorageClass	String	Optional	<b>Storage class</b> specified during the object creation
Body	String or stream.Readable	Optional	Content of the object to be uploaded. Both character strings and instances of <b>stream.Readable</b> 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 <b>SourceFile</b> is set and its default value is <b>0</b> .
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
SuccessActionRedirect	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 <b>SourceFile</b> 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: <ul style="list-style-type: none"> <li>kms</li> </ul>
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be <b>null</b> .
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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 <b>null</b> .
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

```
obsClient.appendObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  // SourceFile : 'localfile',
  Body : 'Hello OBS',
  Position: 0
},(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('NextPosition-->' + result.InterfaceResult.NextPosition);
    }else{
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

## 5.3 GET Object

### API Description

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

## Method Definition

ObsClient.getObject

## Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
VersionId	String	Optional	Object version ID
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 <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .
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 <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .
Range	String	Optional	Download range. The value range is [0, object length-1] and is in the format of <b>bytes=x-y</b> . The maximum length of <b>Range</b> is the length of the object minus 1. If it exceeds this value, the length of the object minus 1 is used.

Field	Type	Optional or Mandatory	Description
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 <b>Cache-Control</b> header in the response.
ResponseContentDisposition	String	Optional	Rewrites the <b>Content-Disposition</b> header in the response.
ResponseContentEncoding	String	Optional	Rewrites the <b>Content-Encoding</b> header in the response.
ResponseContentLanguage	String	Optional	Rewrites the <b>Content-Language</b> header in the response.
ResponseContentType	String	Optional	Rewrites the <b>Content-Type</b> header in the response.
ResponseExpires	String	Optional	Rewrites the <b>Expires</b> header in the response
ImageProcess	String	Optional	Image processing parameter
SaveAsFile	String	Optional	Target path to which the object is downloaded (containing the file name)
SaveAsStream	Boolean	Optional	Whether the object is returned as an instance of <b>stream.Readable</b>
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: <ul style="list-style-type: none"> <li>• AES256</li> </ul>
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	<b>Cache-Control</b> header in the response
ContentDisposition	String	<b>Content-Disposition</b> header in the response
ContentEncoding	String	<b>Content-Encoding</b> header in the response
ContentLanguage	String	<b>Content-Language</b> header in the response
ContentType	String	MIME type of the object
Expires	String	<b>Expires</b> 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 <b>Origin</b> in the request meets the CORS rules of the bucket, <b>AllowedOrigin</b> in the CORS rules is returned.
AllowHeader	String	If <b>RequestHeader</b> in the request meets the CORS rules of the bucket, <b>AllowedHeader</b> in the CORS rules is returned.
AllowMethod	String	<b>AllowedMethod</b> in the CORS rules of the bucket
ExposeHeader	String	<b>ExposeHeader</b> in the CORS rules of the bucket

Field	Type	Description
MaxAgeSeconds	String	<b>MaxAgeSeconds</b> 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 <b>null</b> when <b>SaveAsFile</b> is set. The content is an instance of <b>stream.Readable</b> when <b>SaveAsStream</b> is set to <b>true</b> . The content is an instance of <b>Buffer</b> when neither <b>SaveAsFile</b> nor <b>SaveAsStream</b> is set.
Metadata	Object	Customized metadata of the object

## Sample Code

```

obsClient.getObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  Range : 'bytes=0-10',
  // SaveAsFile: 'localfile',
  // SaveAsStream: true
},(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('DeleteMarker-->' + result.InterfaceResult.DeleteMarker);
      console.log('LastModified-->' + result.InterfaceResult.LastModified);
      console.log('StorageClass-->' + result.InterfaceResult.StorageClass);
      console.log('Content-->' + result.InterfaceResult.Content.toString());
      console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
    }else{
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});

```

## 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	<b>Pre-defined access control policy</b> specified during object copy
StorageClass	String	Optional	<b>Storage class</b> 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 <i>SourceBucketName/SourceObjectName?versionId=SourceObjectVersionId</i> .
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 <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .

Field	Type	Optional or Mandatory	Description
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 <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .
CacheControl	String	Optional	When an object is copied, this parameter rewrites the <b>Cache-Control</b> header in the response.
ContentDisposition	String	Optional	When an object is copied, this parameter rewrites the <b>Content-Disposition</b> header in the response.
ContentEncoding	String	Optional	When an object is copied, this parameter rewrites the <b>Content-Encoding</b> header in the response.
ContentLanguage	String	Optional	When an object is copied, this parameter rewrites the <b>Content-Language</b> header in the response.
ContentType	String	Optional	When an object is copied, this parameter rewrites the <b>Content-Type</b> header in the response.
Expires	String	Optional	When an object is copied, this parameter rewrites the <b>Expires</b> header in the response.
MetadataDirective	String	Optional	<b>Replication policy</b>
Metadata	Object	Optional	Customized metadata of the target object
WebsiteRedirectLocation	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
SuccessActionRedirect	String	Optional	Redirection address after the copy is successful

Field	Type	Optional or Mandatory	Description
SseKms	String	Optional	Algorithm used to encrypt a target object in SSE-KMS mode. The value can be: <ul style="list-style-type: none"> <li>kms</li> </ul>
SseKmsKey	String	Optional	Master key used to encrypt a target object in SSE-KMS encryption, which can be left <b>null</b>
SseC	String	Optional	Algorithm used to encrypt a target object in SSE-C mode. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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

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

## Returned Result (InterfaceResult)

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

Field	Type	Description
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('Deleted:');
      for(let i=0;i<result.InterfaceResult.Deleted.length;i++){
        console.log('Deleted[' + i + ':]');
        console.log('Key-->' + result.InterfaceResult.Deleted[i]['Key']);
        console.log('VersionId-->' + result.InterfaceResult.Deleted[i]['VersionId']);
        console.log('DeleteMarker-->' + result.InterfaceResult.Deleted[i]['DeleteMarker']);
        console.log('DeleteMarkerVersionId-->' + result.InterfaceResult.Deleted[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.
RequestHeader	String	Optional	HTTP headers in the cross-domain request
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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 <b>null</b> .

Field	Type	Description
Restore	String	Restore status of the object in the OBS Archive storage class
AllowOrigin	String	If <b>Origin</b> in the request meets the CORS rules of the bucket, <b>AllowedOrigin</b> in the CORS rules is returned.
AllowHeader	String	If <b>AccessControlRequestHeaders</b> in the request meets the CORS rules of the bucket, <b>AllowedHeader</b> in the CORS rules is returned.
AllowMethod	String	<b>AllowedMethod</b> in the CORS rules of the bucket
ExposeHeader	String	<b>ExposeHeader</b> in the CORS rules of the bucket
MaxAgeSeconds	String	<b>MaxAgeSeconds</b> 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('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 <b>REPLACE_NEW</b> or <b>REPLACE</b> . <b>REPLACE_NEW</b> : 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. <b>REPLACE</b> : Use the header field carried in the current request to replace the original metadata. The metadata that is not specified (except <b>StorageClass</b> ) will be deleted.

Field	Type	Optional or Mandatory	Description
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.
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 <b>STANDARD</b> , <b>WARM</b> , and <b>COLD</b> . The spell of these values is case sensitive.
WebsiteRedirectLocation	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

Field	Type	Description
MetadataDirective	String	Metadata operation indicator. The value can be <b>REPLACE_NEW</b> or <b>REPLACE</b> .
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
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 <b>null</b> .
Content-Language	String	If <b>Origin</b> in the request meets the CORS rules of the bucket, <b>AllowedOrigin</b> in the CORS rules is returned.
Expires	String	If <b>AccessControlRequestHeaders</b> in the request meets the CORS rules of the bucket, <b>AllowedHeader</b> in the CORS rules is returned.
Metadata	Object	Customized metadata of the object

## Sample Code

```
obsClient.setObjectMetadata({
  Bucket:'bucketname',
  Key : 'objectkey',
  CacheControl: 'CacheControl',
  ContentDisposition: 'ContentDisposition',
  ContentLanguage: 'ContentLanguage',
  ContentEncoding: 'ContentEncoding',
  ContentType: 'ContentType',
  StorageClass: 'StorageClass',
  Expires: 'Expires',
  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('CacheControl-->' + result.InterfaceResult.CacheControl);
      console.log('ContentDisposition-->' + result.InterfaceResult.ContentDisposition);
      console.log('ContentLanguage-->' + result.InterfaceResult.ContentLanguage);
      console.log('Metadata-->' + result.InterfaceResult.Metadata);
      console.log('Expires-->' + result.InterfaceResult.Expires);
    }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	<a href="#">Pre-defined ACL</a>
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	<a href="#">Available Grantee Types</a>

Field		Type	Optional or Mandatory	Description
	ID	String	If <b>Type</b> is <b>CanonicalUser</b> , this field is optional. If <b>Type</b> is <b>Group</b> , this field must be <b>null</b> .	ID of the domain to which the grantee belongs
	URI	String	If <b>Type</b> is <b>Group</b> , this field is mandatory. If <b>Type</b> is <b>CanonicalUser</b> , this field must be <b>null</b> .	<b>Available Grantee Groups</b>
	Permission	String	Mandatory	<b>Granted permission</b>

 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',
  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.GroupAllUsers},Permission:obsClient.enums.PermissionWrite},
    {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);
    }
  }
});
```

## 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

Field		Type	Description
VersionId		String	Object version ID
Owner		Object	Object owner
	ID	String	ID of the domain to which the object owner belongs
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 <b>Type</b> of <b>Grantee</b> is <b>Group</b> .
	URI	String	Grantee group. This field is null when <b>Type</b> of <b>Grantee</b> is <b>CanonicalUser</b> .
	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	<a href="#">Pre-defined access control policy</a>
StorageClass	String	Optional	<a href="#">Storage class</a> 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: <ul style="list-style-type: none"> <li>kms</li> </ul>
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be <b>null</b> .
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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 <b>stream.Readable</b> 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 <b>SourceFile</b> is set and its default value is <b>0</b> .
PartSize	Number	Optional	Size (in bytes) of a part in the source file. This field is effective when <b>SourceFile</b> is set and its default value is file size minus <b>Offset</b> . 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: <ul style="list-style-type: none"> <li>• AES256</li> </ul>
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 <i>SourceBucketName/SourceObjectName?versionId=SourceObjectVersionId</i> .
CopySourceRange	String	Optional	Copy source range. The value range is [0, source object length-1] and is in the format of <i>bytes=x-y</i> . If the maximum length of <b>CopySourceRange</b> 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: <ul style="list-style-type: none"> <li>• AES256</li> </ul>
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: <ul style="list-style-type: none"> <li>• AES256</li> </ul>
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 <b>true</b> , not all multipart uploads are returned. If the field value is <b>false</b> , 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 POST Object Restore

### 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	<a href="#">Restore option</a>

## Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
RestoreStatus	String	Restoration status of the object. <b>AVAILABLE</b> indicates that the object can be downloaded. <b>INPROGRESS</b> 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: <ul style="list-style-type: none"><li>• GET</li><li>• POST</li><li>• PUT</li><li>• DELETE</li><li>• HEAD</li></ul>
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: <ul style="list-style-type: none"> <li>• versions</li> <li>• uploads</li> <li>• location</li> <li>• storageinfo</li> <li>• quota</li> <li>• storagePolicy</li> <li>• acl</li> <li>• append</li> <li>• logging</li> <li>• policy</li> <li>• lifecycle</li> <li>• website</li> <li>• versioning</li> <li>• cors</li> <li>• notification</li> <li>• tagging</li> <li>• delete</li> <li>• restore</li> </ul>
Expires	Number	Optional	Expiration time of the signed URL, in seconds. The default value is <b>300</b> .
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 <b>key</b> field in the form
Expires	Number	Optional	Expiration time of the browser-based authentication (unit: seconds). The default value is <b>300</b> .
FormParams	Object	Optional	Other parameters of the browser-based upload except for <b>key</b> , <b>policy</b> , and <b>signature</b> . Possible values are: <ul style="list-style-type: none"> <li>• acl</li> <li>• cache-control</li> <li>• content-type</li> <li>• content-disposition</li> <li>• content-encoding</li> <li>• expires</li> </ul>

## Returned Result

Field	Type	Description
OriginPolicy	String	<b>policy</b> not encoded by Base64. This parameter can only be used for verification.
Policy	String	<b>policy</b> in a form
Signature	String	<b>signature</b> 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 defaults to 5 MB.
TaskNum	Number	Optional	Maximum number of threads that can be concurrently executed for upload. The default value is <b>20</b> .
EnableCheckpoint	Boolean	Optional	Whether to enable the resumable upload mode. The default value is <b>false</b> , 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 <b>null</b> , 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 <b>false</b> , which indicates that the content will not be verified.

Field	Type	Optional or Mandatory	Description
ContentType	String	Optional	MIME type of the object
ACL	String	Optional	<a href="#">Pre-defined access control policy</a>
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: <ul style="list-style-type: none"> <li>kms</li> </ul>
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. This field can be <b>null</b> .
sseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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

Field	Type	Description
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.uploadFile({
  Bucket : 'bucketname',
  Key : 'objectkey',
  UploadFile : 'localfile',
  PartSize : 10 * 1024 * 1024,
  EnableCheckpoint : true
}, (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);
    }
  }
});
```

## 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

Field	Type	Optional or Mandatory	Description
DownloadFile	String	Optional	Local path to which the object is downloaded. If this parameter is <b>null</b> , 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 defaults to 5 MB.
TaskNum	Number	Optional	Maximum number of threads that can be concurrently executed for download. The default value is <b>20</b> .
EnableCheckpoint	Boolean	Optional	Whether to enable the resumable upload mode. The default value is <b>false</b> , which indicates that this mode is disabled.
CheckpointFile	String	Optional	File used to record the download progress. This parameter is effective only in the resumable download mode. If this parameter is <b>null</b> , 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.

Field	Type	Optional or Mandatory	Description
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: <ul style="list-style-type: none"> <li>AES256</li> </ul>
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
DeleteMarker	String	Whether the deleted object is a delete marker
LastModified	String	Time when the last modification was made to the object
CacheControl	String	<b>Cache-Control</b> header in the response
ContentDisposition	String	<b>Content-Disposition</b> header in the response
ContentEncoding	String	<b>Content-Encoding</b> header in the response
ContentLanguage	String	<b>Content-Language</b> header in the response
ContentType	String	MIME type of the object
Expires	String	<b>Expires</b> 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

Field	Type	Description
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is <b>null</b> .
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

```
obsClient.downloadFile({
  Bucket : 'bucketname',
  Key : 'objectkey',
  DownloadFile : 'localfile',
  PartSize : 10 * 1024 * 1024,
  EnableCheckpoint : true
}, (err, result) => {
  if(err){
    console.error('Error-->' + err);
  }else{
    console.log('RequestId-->' + result.InterfaceResult.RequestId);
    console.log('LastModified-->' + result.InterfaceResult.LastModified);
    console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
  }
});
```

# A Change History

---

Release Date	What's New
2019-03-30	This is the seventh official release. Added the following section: <a href="#">Overview</a>
2019-03-05	This is the sixth official release. Modified the following sections: Added parameters and description related to FunctionGraph in sections <a href="#">PUT Bucket notification</a> and <a href="#">GET Bucket notification</a> .
2019-01-17	This is the fifth official release. Modified the code example in section <a href="#">Initializing an Instance of ObsClient</a> .
2018-08-31	This is the fourth official release. Added the following sections: <a href="#">Append Object</a> <a href="#">Generating Browser-Based Upload Parameters with Authentication Information</a>

Release Date	What's New
2018-03-31	<p>This issue is the third official release.</p> <p>Added the following sections:</p> <ul style="list-style-type: none"><li><a href="#">Pre-defined Constants</a></li><li><a href="#">Set Bucket storagePolicy</a></li><li><a href="#">GET Bucket storagePolicy</a></li></ul> <p>Modified the following sections:</p> <p>Modified returned result parameters and the sample code in <a href="#">Creating a Signed URL</a>.</p> <p>Added the <b>Transitions</b> and <b>NoncurrentVersionTransitions</b> fields in <a href="#">PUT Bucket lifecycle</a>.</p> <p>Added the <b>Transitions</b> and <b>NoncurrentVersionTransitions</b> fields in <a href="#">GET Bucket lifecycle</a>.</p> <p>Added the <b>StorageClass</b> field in <a href="#">PUT Object</a>.</p> <p>Added the <b>ImageProcess</b> field in <a href="#">GET Object</a>.</p> <p>Added the <b>StorageClass</b> field in <a href="#">PUT Object - Copy</a>.</p> <p>Added the <b>StorageClass</b> field in <a href="#">Initiate Multipart Upload</a>.</p>
2017-11-30	<p>This issue is the second official release.</p> <p>Added the following sections:</p> <ul style="list-style-type: none"><li><a href="#">Performing a Resumable Upload</a></li><li><a href="#">Performing a Resumable Download</a></li></ul>
2017-10-31	<p>This issue is the first official release.</p>