

Object Storage Service

BrowserJS SDK API Reference

Issue 01
Date 2025-12-04



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

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

Trademarks and Permissions



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

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

Notice

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

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

Contents

1 Overview	1
2 Initialization	2
2.1 Initializing an Instance of ObsClient	2
2.2 Initializing Logs	3
2.3 Request Object	4
2.4 SDK Common Result Objects	5
2.5 Result Returned via a Callback Function	6
2.6 Result Returned via the Promise Object	6
3 Pre-defined Constants	8
3.1 Permission Types	8
3.2 Available Grantee Types	9
3.3 Available Grantee Groups	9
3.4 Pre-defined Access Control Policies	9
3.5 Storage Classes	10
3.6 Restore Options	10
3.7 Metadata Replication Policy	10
4 Bucket-Related APIs	11
4.1 Before You Start	11
4.2 HEAD Bucket	11
4.3 DELETE Bucket	12
4.4 GET Objects	13
4.5 GET Object versions	16
4.6 List Multipart uploads	20
4.7 Obtain Bucket Metadata	23
4.8 GET Bucket location	24
4.9 GET Bucket storageinfo	25
4.10 PUT Bucket quota	26
4.11 GET Bucket quota	27
4.12 Set Bucket storagePolicy	28
4.13 GET Bucket storagePolicy	29
4.14 PUT Bucket acl	30
4.15 GET Bucket acl	32

4.16 PUT Bucket logging.....	34
4.17 GET Bucket logging.....	36
4.18 PUT Bucket policy.....	37
4.19 GET Bucket policy.....	39
4.20 DELETE Bucket policy.....	40
4.21 PUT Bucket lifecycle.....	41
4.22 GET Bucket lifecycle.....	44
4.23 DELETE Bucket lifecycle.....	46
4.24 PUT Bucket website.....	47
4.25 GET Bucket website.....	50
4.26 DELETE Bucket website.....	52
4.27 PUT Bucket versioning.....	53
4.28 GET Bucket versioning.....	54
4.29 GET Bucket cors.....	55
4.30 DELETE Bucket cors.....	57
4.31 PUT Bucket tagging.....	58
4.32 GET Bucket tagging.....	59
4.33 DELETE Bucket tagging.....	61
5 Object-Related APIs.....	62
5.1 Before You Start.....	62
5.2 PUT Object.....	63
5.3 Append Object.....	66
5.4 GET Object.....	68
5.5 PUT Object - Copy.....	72
5.6 DELETE Object.....	76
5.7 DELETE Objects.....	77
5.8 Obtain Object Metadata.....	79
5.9 PUT Object acl.....	81
5.10 GET Object acl.....	83
5.11 Initiate Multipart Upload.....	85
5.12 PUT Part.....	87
5.13 PUT Part - Copy.....	89
5.14 List Parts.....	91
5.15 Complete Multipart Upload.....	94
5.16 DELETE Multipart upload.....	95
5.17 Restoring Archive Objects.....	96
6 Other APIs.....	99
6.1 Creating a Signed URL.....	99
6.2 Generating Browser-Based Upload Parameters with Authentication Information.....	101
6.3 Upload File (Resumable Upload).....	103
A Change History.....	108

1 Overview

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

To learn the complete procedures for using the OBS BrowserJS SDK (including installation, initialization, development), as well as FAQs, scenarios, and code examples, see the [Object Storage Service BrowserJS SDK Developer Guide](#).

If you have any questions during the development, post them on GitHub.

2 Initialization

2.1 Initializing an Instance of ObsClient

API Description

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

Method Definition

1. Constructor form: `ObsClient(parameter)`
2. Factory 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> . To view the endpoints available for OBS, see Regions and Endpoints .

Field	Type	Optional or Mandatory	Description
timeout	Number	Optional	The total timeout period (in seconds) of an HTTP/HTTPS request. The default value is 300 . If the network is unstable or the size of the file to be uploaded is large, you are advised to set a larger value for timeout .
is_cname	Boolean	Optional	Whether to use self-defined domain name to access OBS. The default value is false .
useRawXhr	Boolean	Optional	Whether to use the native XHR to send Ajax requests. The default value is false .

Sample Code

```
// Create an instance of ObsClient.
var obsClient = new ObsClient({
  // Hard-coded or plaintext AK/SK are risky. For security purposes, encrypt your AK/SK and store them in
  // the configuration file or environment variables. In this example, the AK/SK are stored in environment
  // variables for identity authentication. Before running this example, configure environment variables
  // AccessKeyID and SecretAccessKey.
  // The front-end code does not have the process environment variable, so you need to use a module
  // bundler like webpack to define the process variable.
  // Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca\_01\_0003.html.
  access_key_id: process.env.AccessKeyID,
  secret_access_key: process.env.SecretAccessKey,
  timeout : 300,
  // Replace the example endpoint with the actual one in your case.
  server: 'https://obs.ap-southeast-1.myhuaweicloud.com'
});
```

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
level	String	Mandatory	Log level. Possible values are debug , info , warn , and error .

Sample Code

```
obsClient.initLog({  
  level:'info'  
});
```

2.3 Request Object

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 **Bucket** field contained in the request object is used to specify the bucket name. For an object-related API, the **Bucket** field and **Key** field contained in the request object are used 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Other parameters	For details, see sections "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 HTTP status code and error code.
-	Status	Number	HTTP status code. If the value is smaller than 300 , the operation succeeds. Otherwise, the operation fails.
	Code	String	Error code returned by the OBS server. If Status is smaller than 300 , the value is null.
	Message	String	Error description returned by the OBS server. If Status is smaller than 300 , the value is null.
	HostId	String	Requested server ID. If Status is smaller than 300 , the value is null.
	RequestId	String	Request ID returned by the OBS server
	Id2	String	Request ID2 returned by the OBS server
	Indicator	String	Detailed error code returned by the OBS server. If Status is smaller than 300 , the value is null.
InterfaceResult		Object	Result data generated after the operation is successful. If Status is greater than 300 , the value is null.
-	RequestId	String	Request ID returned by the OBS server
	Id2	String	Request ID2 returned by the OBS server
	Other fields	For details, see "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

This example uploads string **Hello OBS** to bucket **bucketname** as object **objectkey**.

The example code is as follows:

```
obsClient.putObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  Body : 'Hello OBS'
}, function(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

This example uploads string **Hello OBS** to bucket **bucketname** as object **objectkey**.

The example code is as follows:

```
obsClient.putObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  Body : 'Hello OBS'
}).then(function(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(function(err) {
  console.error('Error-->' + err);
});
```

3 Pre-defined Constants

3.1 Permission Types

Access Method	Type	Description
ObsClient.enums.PermissionRead	String	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, bucket metadata, and object versions in the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient.enums.PermissionWrite	String	A grantee with this permission for a bucket can upload, overwrite, and delete any object or part in the bucket. This permission is not applicable to objects.
ObsClient.enums.PermissionReadAcp	String	A grantee with this permission can obtain the ACL of a bucket or object. A bucket or object owner has this permission permanently.
ObsClient.enums.PermissionWriteAcp	String	A grantee with this permission can update the ACL of a bucket or object. A bucket or object owner has this permission permanently. A grantee with this permission can modify the access control policy and thus the grantee obtains full access permissions.

Access Method	Type	Description
ObsClient.enums.PermissionFullControl	String	A grantee with this permission for a bucket has PermissionRead, PermissionWrite, PermissionReadAcp, and PermissionWriteAcp permissions for the bucket. A grantee with this permission for an object has PermissionRead, PermissionReadAcp, and PermissionWriteAcp permissions for the object.

3.2 Available Grantee Types

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

3.3 Available Grantee Groups

Access Method	Type	Description
ObsClient.enums.GroupAllUsers	String	All users
ObsClient.enums.GroupAuthenticatedUsers	String	Authorized users. This constant is deprecated.
ObsClient.enums.GroupLogDelivery	String	Log delivery group. This constant is deprecated.

3.4 Pre-defined Access Control Policies

Access Method	Type	Description
ObsClient.enums.AclPrivate	String	Private read/write
ObsClient.enums.AclPublicRead	String	Public read
ObsClient.enums.AclPublicReadWrite	String	Public read/write

Access Method	Type	Description
ObsClient.enums.AclPublicReadDelivered	String	public read on a bucket as well as objects in the bucket
ObsClient.enums.AclPublicReadWriteDelivered	String	Public read/write on a bucket as well as objects in the bucket

3.5 Storage Classes

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

3.6 Restore Options

Access Method	Type	Description
ObsClient.enums.RestoreTierExpedited	String	Objects can be restored at an expedited speed within 1 to 5 minutes.
ObsClient.enums.RestoreTierStandard	String	Objects can be restored at a standard speed within 3 to 5 hours.

3.7 Metadata Replication Policy

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

4 Bucket-Related APIs

4.1 Before You Start

Due to browser limitations, configure CORS for buckets before calling bucket-related APIs as follows:

- Set **AllowedOrigin** to *****.
- Set **AllowedMethod** to **PUT, GET, POST, DELETE, and HEAD**.
- Set **AllowedHeader** to *****.
- Set **ExposeHeader** to:
 - ETag
 - x-obs-request-id
 - x-obs-id-2
 - x-reserved-indicator
 - x-obs-api
 - x-obs-bucket-location
 - x-obs-version
 - x-obs-storage-class
 - x-default-storage-class

NOTE

Bucket-related API functions of **ObsClient** are case insensitive. For example, **ObsClient.getBucketLocation** and **ObsClient.GetBucketLocation** indicate the same function.

4.2 HEAD Bucket

API Description

You can use this API to check whether a bucket exists. If the returned HTTP status code is **200**, the bucket exists. If the returned HTTP status 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.headBucket({
  Bucket : 'bucketname'
}, function (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');
    }
  }
});
```

4.3 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucket({
  Bucket : 'bucketname'
}, function (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);
    }
  }
});
```

4.4 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Prefix	String	Optional	Name prefix that the objects to be listed must contain
Marker	String	Optional	Object name to start with when listing objects in a bucket. All objects are listed in the lexicographical order.
MaxKeys	Number	Optional	Maximum number of objects returned in the response. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.
Delimiter	String	Optional	Character used to group object names. If the object name contains the Delimiter parameter, the character string from the first character to the first delimiter in the object name is grouped under a single result element, CommonPrefix . (If a prefix is specified in the request, the prefix must be removed from the object name.)

Returned Result (InterfaceResult)

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

Field		Type	Description
Delimiter		String	Character used to group versioning object names, which is consistent with that set in the request
IsTruncated		String	Whether all versioning objects are returned. If the field value is true , not all versioning objects are returned. If the field value is false , all versioning objects are returned.
Prefix		String	Name prefix of versioning objects, 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
MaxKeys		String	Maximum number of listed objects, which is consistent with that set in the request
Contents		Array	Object list
-	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	ID of the domain to which the object owner belongs
-	StorageClass	String	Storage class of the object
-	Type	String	Whether the object is an appendable object
CommonPrefixes		Array	List of object name prefixes grouped according to the Delimiter parameter (if specified)
-	Prefix	String	Object name prefix grouped according to the Delimiter parameter

Sample Code

```
obsClient.listObjects({
  Bucket : 'bucketname',
  Prefix : 'prefix',
  MaxKeys : 100
},function (err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
    if(result.CommonMsg.Status < 300){
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      for(var 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);
    }
  }
});
```

4.5 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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

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 versioning object names, which is consistent with that set in the request

Field		Type	Description
Prefix		String	Name prefix of versioning objects, which is consistent with that set in the request
IsTruncated		String	Whether all versioning objects are returned. If the field value is true , not all versioning objects are returned. If the field value is false , all versioning objects are returned.
KeyMarker		String	Object name to start with for listing versioning objects, which is consistent with that set in the request
VersionIdMarker		String	Start position for listing 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
NextVersionIdMarker		String	Version ID to start with upon the next request for listing versioning objects. It is used with the NextKeyMarker parameter.
MaxKeys		String	Maximum number of listed versioning objects, which is consistent with that set in the request
Versions		Array	List of versioning objects
-	ETag	String	MD5 value of the object
	Size	String	Object size in bytes
	Key	String	Object name
	VersionId	String	Object version ID
	IsLatest	String	Whether the object is of the latest version. If the field value is true , the object is of the latest version.
	LastModified	String	Time when the last modification was made to the object
	Owner	Object	Object owner
	-	ID	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

Field			Type	Description
DeleteMarkers			Array	List of versioning 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 field value is true , the object is of the latest version.
	LastModified		String	Time when the last modification was made to the object
CommonPrefixes			Array	List of object name prefixes grouped according to the Delimiter parameter (if specified)
-	Prefix		String	Object name prefix grouped according to the Delimiter parameter

Sample Code

```
obsClient.listVersions({
  Bucket : 'bucketname',
  Prefix : 'prefix',
  MaxKeys : 100
},function (err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
    if(result.CommonMsg.Status < 300){
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      console.log('Versions:');
      for(var j=0;j<result.InterfaceResult.Versions;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(var 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);  
    }  
}  
});
```

4.6 List Multipart uploads

API Description

You can use this API to list 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Delimiter	String	Optional	Character used to group object names involved in multipart uploads. If the object name contains the Delimiter parameter, the character string from the first character to the first delimiter in the object name is grouped under a single result element, CommonPrefix . (If a prefix is specified in the request, the prefix must be removed from the object name.)
Prefix	String	Optional	Prefix that the object names in the multipart uploads to be listed must contain
MaxUploads	Number	Optional	Maximum number of returned multipart uploads. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.

Field	Type	Optional or Mandatory	Description
KeyMarker	String	Optional	Object name to start with when listing multipart uploads
UploadIdMarker	String	Optional	Upload ID after which the multipart upload listing begins. It is effective only when used with KeyMarker so that multipart uploads after UploadIdMarker of KeyMarker will be listed.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
Bucket	String	Bucket name
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 is used with the nextKeyMarker parameter.
Delimiter	String	Character used to group object names in multipart uploads, which is consistent with that set in the request
Prefix	String	Object name prefix in multipart uploads, which is consistent with the same parameter in the request
MaxUploads	String	Maximum number of listed multipart uploads, which is consistent with the same parameter in the request

Field		Type	Description
IsTruncated		String	Whether all versioning objects are returned. If the field value is true , not all versioning objects are returned. If the field value is false , all versioning objects 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 initiator belongs
	Owner	Object	Owner of the multipart upload, which is consistent with Initiator
	-	ID	String ID of the domain to which the initiator belongs
	Initiated	String	Time when the multipart upload was initiated
-	StorageClass	String	Storage class of the object to be uploaded
CommonPrefixes		Array	List of object name prefixes grouped according to the Delimiter parameter (if specified)
-	Prefix	String	Object name prefix grouped according to the Delimiter parameter

Sample Code

```
obsClient.listMultipartUploads ({
  Bucket : 'bucketname',
  Prefix : 'prefix',
  MaxUploads : 100
},function (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(var 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('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

4.7 Obtain Bucket Metadata

API Description

You can use this API to send a HEAD request to obtain the bucket metadata.

Method Definition

ObsClient.getBucketMetadata

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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

Sample Code

```
obsClient.getBucketMetadata({
  Bucket: 'bucketname'
},function (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);
      }else{
        console.log('Status-->' + result.CommonMsg.Status);
      }
    }
  }
});
```

4.8 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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'
},function (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);  
    }  
}  
});
```

4.9 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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
StandardSize	String	Size of Standard objects in the bucket
StandardObjectNumber	String	Number of Standard objects in the bucket
WarmSize	String	Size of Infrequent Access objects in the bucket returned for an OBS request

Field	Type	Description
WarmObjectNumber	String	Number of Infrequent Access objects in the bucket returned for an OBS request
ColdSize	String	Size of Archive objects in the bucket returned for an OBS request
ColdObjectNumber	String	Number of Archive objects in the bucket returned for an OBS request
DeepArchiveSize	String	Size of Deep Archive objects in the bucket
DeepArchiveObject-Number	String	Number of Deep Archive objects in the bucket

Sample Code

```
obsClient.getBucketStorageInfo({
  Bucket : 'bucketname'
},function (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);
    }
  }
});
```

4.10 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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
},function (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);
    }
  }
});
```

4.11 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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'
},function (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);
    }
  }
});
```

4.12 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
StorageClasses	String	Mandatory	Storage class of the bucket

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketStoragePolicy({
  Bucket : 'bucketname',
  StorageClass: obsClient.enums.StorageClassWarm
},function (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);
    }
  }
});
```

4.13 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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'
},function (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);
    }
  }
});
```

4.14 PUT Bucket acl

API Description

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

Method Definition

```
ObsClient.setBucketAcl
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			String	Mandatory	Bucket name
RequestDate			String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
ACL			String	Optional	Pre-defined access control policy
Owner			Object	Optional	Bucket owner
-	ID		String	Mandatory	ID of the domain to which the bucket owner belongs
Grants			Array	Optional	List of grantees' permission information
-	Grantee		Object	Mandatory	Grantee
	-	Type	String	Mandatory	Grantee type
		ID	String	Mandatory when Type is CanonicalUser . In other cases, leave it null.	ID of the domain to which the grantee belongs
		URI	String	Mandatory when Type is Group . In other cases, leave it null.	Grantee group

Field		Type	Optional or Mandatory	Description
	Permission	String	Mandatory	Granted permission
	Delivered	Boolean	Optional	Whether an object inherits the ACL of its residing bucket

 NOTE

- **Owner** and **Grants** must be used together and they cannot be used with **ACL**.
- 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}
  ]
},function (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);
    }
  }
});
```

4.15 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
Owner		Object	Bucket owner
-	ID	String	ID of the domain to which the bucket owner belongs
	Name	String	Name of the bucket owner
Grants		Array	List of grantees' permission information
-	Grantee		Grantee
	-	Name	Grantee name. This field is null when Type of Grantee is Group .
		ID	ID of the domain to which the grantee belongs. This field is null when Type of Grantee is Group .
		URI	Grantee group. This field is null when Type of Grantee is CanonicalUser .
	Permission		Granted permission
	Delivered		Whether an object inherits the ACL of its residing bucket

Sample Code

```
obsClient.getBucketAcl ({  
  Bucket : 'bucketname'
```

```
},function (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(var 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);  
    }  
  }  
});
```

4.16 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
RequestDate		String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Agency		String	Mandatory when setting bucket logging	Agency name
LoggingEnabled		Object	Optional	Log configuration information
-	TargetBucket	String	Mandatory	Target bucket for which logs are generated

Field			Type	Optional or Mandatory	Description	
	TargetPrefix		String	Mandatory	Name prefix of a to-be-logged object in the target bucket	
	TargetGrants		Array	Optional	List of grantees' permission information	
	-	Grantee	Object	Optional	Grantee	
		-	Type	String	Mandatory	Grantee type
			ID	String	Mandatory when Type is Canonical User . In other cases, leave it null.	ID of the domain to which the grantee belongs
			URI	String	Mandatory when Type is Group . In other cases, leave it null.	Grantee group
		Permission		String	Optional	Granted permission

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketLogging ({
  Bucket : 'bucketname',
  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}
```

```
    }  
  },function (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);  
      }  
    }  
  }  
});
```

4.17 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Field				Type	Description
	TargetGrants			Array	List of grantees' permission information
	-	Grantee		Object	Grantee
		-	ID	String	ID of the domain to which the grantee belongs. This field is null when Type of Grantee is Group .
			URI	String	Grantee group. This field is null when Type of Grantee is CanonicalUser .
		Permission		String	Granted permission

Sample Code

```
obsClient.getBucketLogging ({
  Bucket : 'bucketname'
},function (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(var i=0;i<result.InterfaceResult.LoggingEnabled.TargetGrants.length;i++){
          console.log('Grant[' + i + ']:');
          console.log('Grantee[ID]-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Grantee']['ID']);
          console.log('Grantee[URI]-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Grantee']['URI']);
          console.log('Permission-->' + result.InterfaceResult.LoggingEnabled.TargetGrants[i]
['Permission']);
        }
      }
    }else{
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

4.18 PUT Bucket policy

API Description

You can use this API to set a 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Policy	String	Mandatory	Policy information in JSON format. For details about the format, see Policy Format .

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketPolicy ({
  Bucket : 'bucketname',
  Policy: 'your policy',
},function (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.

4.19 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.getBucketPolicy ({
  Bucket : 'bucketname'
},function (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);
    }
  }
});
```

4.20 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucketPolicy ({
  Bucket : 'bucketname'
},function (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);
    }
  }
});
```

4.21 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
RequestDate			String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Rules			Array	Mandatory	Lifecycle rules of the bucket
-	Transitions		Array	Optional	List of object transition policies
	-	StorageClass	String	Mandatory	Storage class of the object after transition NOTE The Standard storage class is not supported.

Field			Type	Optional or Mandatory	Description
		Date	String	Mandatory when Days is not set	Date when an object will be transited. The value must conform with the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01T00:00:00Z.
		Days	Number	Mandatory when Date is not set	Number of days after which an object will be transited since its creation. The value must be a positive integer.
-	Expiration		Object	Optional	Expiration time of an object
	-	Date	String	Mandatory when Days is not set	Date when an object expires. The value must conform with the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01T00:00:00Z.
		Days	Number	Mandatory when Date is not set	Number of days after which an object expires since its creation. The value must be a positive integer.
	ID		String	Optional	Rule ID. It is a 1-255 character string.

Field			Type	Optional or Mandatory	Description
	Prefix		String	Mandatory	Object name prefix identifying one or more 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 this rule is enabled. Possible values are: • Enabled • Disabled
-	NoncurrentVersionTransitions		Array	Optional	List of noncurrent object version transition policies
	-	StorageClass	String	Mandatory	Storage class of the noncurrent object version after transition
		NoncurrentDays	Number	Mandatory	Number of days after which an object will be transitioned since it becomes a noncurrent version. The parameter value must be a positive integer.
-	NoncurrentVersionExpiration		Object	Optional	Expiration time of a noncurrent object version

Field			Type	Optional or Mandatory	Description
	-	NoncurrentDays	Number	Mandatory	Number of days after which an object expires since it becomes a noncurrent version. The field 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'}}
  ]
},function (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 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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 identifying one or more objects to which the rule applies.
	Status		String	Whether the rule is enabled
-	NoncurrentVersionTransitions		Array	List of noncurrent object version transition policies

Field			Type	Description
	-	Storage Class	String	Storage class of the noncurrent object version after transition
		NoncurrentDays	String	Number of days after which an object will be transitioned 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'
},function (err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
    if(result.CommonMsg.Status < 300){
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      for(var 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);
    }
  }
});
```

4.23 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucketLifecycle({
  Bucket : 'bucketname'
});function (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);
    }
  }
};
```

4.24 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
RequestDate		String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
RedirectAllRequestsTo		Object	Optional	Redirection rule of all requests
-	HostName	String	Mandatory	Host name used for redirection
	Protocol	String	Optional	Protocol used for redirection. Possible values are: • http (default) • https
ErrorDocument		Object	Optional	Error page settings
-	Key	String	Optional	Page that is returned when a 4XX error occurs
IndexDocument		Object	Optional	Default page settings
-	Suffix	String	Mandatory	Suffix that is appended to a request initiated for a folder. For example, if the suffix is index.html and you request for samplebucket/images/ , the returned data will be the object named images/index.html in the samplebucket bucket. The suffix can neither be null nor contain slashes (/).
RoutingRules		Array	Optional	Redirection rule list
-	Condition	Object	Optional	Matching condition of a redirection rule

Field			Type	Optional or Mandatory	Description
	-	HttpErrorCodeReturnedEquals	String	Optional	HTTP error code to be matched when a redirection rule takes effect
		KeyPrefixEquals	String	Optional	Object name prefix to be matched when a redirection rule takes effect
	Redirect		Object	Mandatory	Details about a redirection request
	-	Protocol	String	Optional	Protocol used for redirection. Possible values are: • http • https
		HostName	String	Optional	Host name used for redirection
		ReplaceKeyPrefixWith	String	Optional	Object name prefix used in the redirection request
		ReplaceKeyWith	String	Optional	Object name used in the redirection request. This parameter cannot be used together with ReplaceKeyPrefixWith .
		HttpRedirectCode	String	Optional	HTTP status code in the response to the redirection request

 NOTE

- **ErrorDocument**, **IndexDocument**, and **RoutingRules** must be used together and they cannot be used with **RedirectAllRequestsTo**.
- When **ErrorDocument**, **IndexDocument**, and **RoutingRules** are used together, **RoutingRules** can be null.
- You must set either these three fields or **RedirectAllRequestsTo**.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketWebsite({
  Bucket : 'bucketname',
  // RedirectAllRequestsTo : {HostName : 'www.example.com', Protocol : 'https'}
  IndexDocument:{Suffix:'index.html'},
  ErrorDocument:{Key:'error.html'},
  RoutingRules:[
    {Condition:{HttpErrorCodeReturnedEquals:'404'},Redirect:
    {Protocol:'http',ReplaceKeyWith:'NotFound.html'}},
    {Condition:{HttpErrorCodeReturnedEquals:'404'},Redirect:
    {Protocol:'https',ReplaceKeyWith:'test.html'}}
  ]
},function (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 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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 that is returned when a 4XX error occurs
IndexDocument			Object	Default page settings
-	Suffix		String	Suffix that is appended to a request initiated for a folder. For example, if the suffix is index.html and you request for samplebucket/images/ , the returned data will be the object named images/index.html in the samplebucket bucket. The suffix can neither be null nor contain slashes (/).
RoutingRules			Array	Redirection rule list
-	Condition		Object	Matching condition 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
	-	Protocol	String	Protocol used for redirection
		HostName	String	Host name used for redirection

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

Sample Code

```
obsClient.getBucketWebsite({
  Bucket : 'bucketname'
},function (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(var i=0;i<result.InterfaceResult.RoutingRules.length;i++){
        console.log('RoutingRule[' + i + ']:');
        var 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);
    }
  }
});
```

4.26 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucketWebsite({
  Bucket : 'bucketname'
}),function (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);
    }
  }
};
```

4.27 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
VersionStatus	String	Mandatory	Versioning status of the bucket. Possible values are: • Enabled • Suspended

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketVersioning({
  Bucket : 'bucketname',
  VersionStatus : 'Enabled'
},function (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 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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'
},function (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);
    }
  }
});
```

4.29 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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. The value must be an integer.
	ExposeHeader	Array of Strings	Additional response headers allowed by the CORS rule

Sample Code

```
obsClient.getBucketCors({
  Bucket : 'bucketname'
}, function(err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
  }
```

```
if(result.CommonMsg.Status < 300){
    console.log('RequestId-->' + result.InterfaceResult.RequestId);
    for(var 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 (var i=0;i<result.InterfaceResult.CorsRules[k]['AllowedMethod'].length;i++){
            console.log('CorsRule[AllowedMethod]['+i+']-->' + result.InterfaceResult.CorsRules[k]
['AllowedMethod'][i]);
        }
        for(var i=0;i< result.InterfaceResult.CorsRules[k]['AllowedOrigin'].length;i++){
            console.log('CorsRule[AllowedOrigin]['+i+']-->' + result.InterfaceResult.CorsRules[k]
['AllowedOrigin'][i]);
        }
        for(var i=0;i<result.InterfaceResult.CorsRules[k]['AllowedHeader'].length;i++){
            console.log('CorsRule[AllowedHeader]['+i+']-->' + result.InterfaceResult.CorsRules[k]
['AllowedHeader'][i]);
        }
        for(var 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);
}
};
```

4.30 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucketCors({
  Bucket : 'bucketname'
}, function (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 PUT Bucket tagging

API Description

You can use this API to set bucket tags.

Method Definition

```
ObsClient.setBucketTagging
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Tags	Array	Mandatory	Bucket tag set

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

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.setBucketTagging({
  Bucket : 'bucketname',
  Tags : [{Key:'tag1', Value : 'value1'}, {Key:'tag2', Value : 'value2'}]
},function (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);
    }
  }
});
```

4.32 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

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 can contain up to 36 characters
	Value	String	Tag value, which can contain up to 43 characters

Sample Code

```
obsClient.getBucketTagging({
  Bucket : 'bucketname'
},function (err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
    if(result.CommonMsg.Status < 300){
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      for(var i=0;i<result.InterfaceResult.Tags.length;i++){
        var tag = result.InterfaceResult.Tags[i];
        console.log('Tag-->' + tag.Key + ':' + tag.Value);
      }
    }else{
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

4.33 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.deleteBucketTagging({
  Bucket: 'bucketname'
},function (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 Before You Start

Due to browser limitations, configure CORS for buckets before calling object-related APIs as follows:

- Set **AllowedOrigin** to *****.
- Set **AllowedMethod** to **PUT, GET, POST, DELETE, HEAD**.
- Set **AllowedHeader** to *****.
- Set **ExposeHeader** to:
 - ETag
 - Content-Type
 - Content-Length
 - Cache-Control
 - Content-Disposition
 - Content-Encoding
 - Content-Language
 - Expires
 - x-obs-request-id
 - x-obs-id-2
 - x-reserved-indicator
 - x-obs-api
 - x-obs-version-id
 - x-obs-copy-source-version-id
 - x-obs-storage-class
 - x-obs-delete-marker
 - x-obs-expiration
 - x-obs-website-redirect-location
 - x-obs-restore

- x-obs-version
- x-obs-object-type
- x-obs-next-append-position

 NOTE

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

5.2 PUT Object

API Description

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

Method Definition

ObsClient.putObject

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name
Key	String	Mandatory	Object name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
ACL	String	Optional	Pre-defined access control policy specified during object creation
StorageClass	String	Optional	Storage class , which can be specified during the object creation
Body	String	Optional	Object content to be uploaded
SourceFile	File or Blob	Optional	File to be uploaded (The browser must support FileReader .)

Field	Type	Optional or Mandatory	Description
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: Number of uploaded bytes, total bytes, and used time (unit: second).
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. The default value is 0 . This field is effective only when SourceFile is set.
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting
Expires	Number	Optional	Expiration time of the object, in days
SuccessActionRe-direct	String	Optional	Redirection address after the upload is successful
ContentType	String	Optional	MIME type of the object
ContentLength	Number	Optional	Object size in bytes. This parameter is effective when SourceFile is set.
ContentMD5	String	Optional	Base64-encoded MD5 value of the object data to be uploaded. It is provided for the OBS server to verify data integrity.
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: • kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. The value can be null.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: • AES256
SseCKey	String	Optional	Key used in SSE-C encryption. It is calculated by using AES-256.

 NOTE

- **Body** and **SourceFile** cannot be used together.
- If both **Body** and **SourceFile** are null, the size of the object to be uploaded 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
SseKmsKey	String	ID of the KMS master key when SSE-KMS is used
SseC	String	Algorithm used in SSE-C
SseCKeyMd5	String	MD5 value of the key for encrypting objects when SSE-C is used. This value is used to check whether any error occurs during the transmission of the key.

Sample Code

```
obsClient.putObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  Metadata:{meta1:'value1', meta2:'value2'},
  Body : 'Hello OBS',
  ContentType: 'text/plain'
},function (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);
    }
  }
});
```

5.3 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Position	Number	Mandatory	Start position for next appending. This parameter must be set to 0 when you create an appendable object.
ACL	String	Optional	Pre-defined access control policy specified during object creation
StorageClass	String	Optional	Storage class , which can be specified during the object creation
Body	String	Optional	Object content to be uploaded
SourceFile	File or Blob	Optional	File to be uploaded (The browser must support FileReader .)

Field	Type	Optional or Mandatory	Description
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: Number of uploaded bytes, total bytes, and used time (unit: second).
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. The default value is 0 . This field is effective only when SourceFile is set.
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirectLocation	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 size in bytes. This parameter is effective when SourceFile is set.
ContentMD5	String	Optional	Base64-encoded MD5 value of the object data to be uploaded. It is provided for the OBS server to verify data integrity.
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. The value can be null.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: AES256
SseCKey	Buffer	Optional	Key used in SSE-C encryption. It is calculated by using AES-256.

 NOTE

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

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	Object ETag
NextPosition	String	Start position for next appending
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is null.

Sample Code

```
obsClient.appendObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  // SourceFile : 'localfile',
  Body : 'Hello OBS',
  Position: 0
}, function (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);
    }
  }
});
```

5.4 GET Object

API Description

You can use this API to download an object from 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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. The value must conform to the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt .
IfNoneMatch	String	Optional	Returns the source object if its ETag is different from the one specified by this parameter; otherwise, an error code is returned.
IfUnmodifiedSince	String	Optional	Returns the object if it remains unchanged since the time specified by this parameter; otherwise, an exception is thrown. The value must conform to the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt .
Range	String	Optional	Download range. The value range is [0, object length-1] and is in the format of bytes = x-y. The maximum length of Range is the length of the object minus 1. If it exceeds this value, the length of the object minus 1 is used.

Field	Type	Optional or Mandatory	Description
Origin	String	Optional	Origin of the cross-origin request specified by the preflight request. Generally, it is a domain name.
RequestHeader	String	Optional	HTTP header in a cross-domain request
ResponseCacheControl	String	Optional	Rewrites the Cache-Control header in the response.
ResponseContentDisposition	String	Optional	Rewrites the Content-Disposition header in the response.
ResponseContentEncoding	String	Optional	Rewrites the Content-Encoding header in the response.
ResponseContentLanguage	String	Optional	Rewrites the Content-Language header in the response.
ResponseContentType	String	Optional	Rewrites the Content-Type header in the response.
ResponseExpires	String	Optional	Rewrites the Expires header in the response.
ImageProcess	String	Optional	Image processing parameter
SaveByType	String	Optional	Storage class of the object. Possible values are: • text • arraybuffer • blob • file
ProgressCallback	Function	Optional	Callback function for obtaining the download progress NOTE This callback function contains the following parameters in sequence: Number of downloaded bytes, total bytes, and used time (unit: second).
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: • AES256
SseCKey	String	Optional	Key used in SSE-C decryption, which is calculated by using AES-256.

 NOTE

- If the download request includes **IfUnmodifiedSince** or **IfMatch** and **IfUnmodifiedSince** or **IfMatch** is not met, an exception will be thrown with HTTP status code **412 Precondition Failed**.
- If the download request includes **IfModifiedSince** or **IfNoneMatch** and **IfModifiedSince** or **IfNoneMatch** is not met, an exception will be thrown with HTTP status code **304 Not Modified**.

Returned Result (InterfaceResult)

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

Field	Type	Description
Content	String or ArrayBuffer or Blob or Object	Object content. If the SavaByType parameter is not set or is set to text , the value of this parameter is a String object. If the SavaByType parameter is set to arraybuffer , the value of this parameter is an ArrayBuffer object. If the SavaByType parameter is set to blob , the value of this parameter is a Blob object. If the SavaByType parameter is set to file , the value is an Object object, representing the URL of the object (URL validity period: 3600 seconds).
Metadata	Object	Customized metadata of the object You need to add the additional headers allowed to be carried in the response in the CORS configurations. For example, you can add the x-amz-meta-property1 header to obtain customized metadata property1 .

Sample Code

```
obsClient.getObject({
  Bucket : 'bucketname',
  Key : 'objectkey',
  Range: 'bytes=0-10',
  SaveByType: 'file'
},function (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);
      console.log('Metadata-->' + JSON.stringify(result.InterfaceResult.Metadata));
    }else{
      console.log('Code-->' + result.CommonMsg.Code);
      console.log('Message-->' + result.CommonMsg.Message);
    }
  }
});
```

5.5 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
ACL	String	Optional	Pre-defined access control policy specified during object creation
StorageClass	String	Optional	Storage class , which can be specified during the object creation
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. The value must conform to the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt .
CopySourceIfNoneMatch	String	Optional	Copies the source object if its ETag is different from the one specified by this parameter; otherwise, an error code is returned.

Field	Type	Optional or Mandatory	Description
CopySourceIfUnmodifiedSince	String	Optional	Copies the source object if it is changed before the time specified by this parameter; otherwise, an error code is returned. The value must conform to the HTTP time format specified in http://www.ietf.org/rfc/rfc2616.txt .
CacheControl	String	Optional	Rewrites the Cache-Control header in the response.
ContentDisposition	String	Optional	Rewrites the Content-Disposition header in the response.
ContentEncoding	String	Optional	Rewrites the Content-Encoding header in the response.
ContentLanguage	String	Optional	Rewrites the Content-Language header in the response.
ContentType	String	Optional	Rewrites the Content-Type header in the response.
Expires	String	Optional	Rewrites the Expires header in the response.
MetadataDirective	String	Optional	Replication policy
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.
SseKms	String	Optional	Algorithm used to encrypt the target object in SSE-KMS mode. The value can be: kms
SseKmsKey	String	Optional	Master key used to encrypt the target object in SSE-KMS mode. The value can be null.
SseC	String	Optional	Algorithm used to encrypt the target object in SSE-C mode. The value can be: AES256

Field	Type	Optional or Mandatory	Description
SseCKey	String	Optional	Key used to encrypt the target object in SSE-C mode, which is calculated by using AES-256
CopySourceSseC	String	Optional	Algorithm used to decrypt the source object in SSE-C mode. The value can be: • AES256
CopySourceSseCKey	String	Optional	Key used to decrypt the source object in SSE-C mode, which is calculated by using AES-256

 NOTE

- If the object copy request includes **CopySourceIfUnmodifiedSince**, **CopySourceIfMatch**, **CopySourceIfModifiedSince**, or **CopySourceIfNoneMatch**, and the specified condition is not met, an exception will be thrown with HTTP status code **412 Precondition Failed** returned.
- **CopySourceIfModifiedSince** and **CopySourceIfNoneMatch** can be used together. So do **CopySourceIfUnmodifiedSince** and **CopySourceIfMatch**.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag of the target object
LastModified	String	Time when the last modification was made to the target object
VersionId	String	Version ID of the target object. This field is null if versioning is not enabled for the target bucket.
CopySourceVersionId	String	Version ID of the source object. This field is null if versioning is not enabled for the source bucket.

Sample Code

```
obsClient.copyObject({
  Bucket:'bucketname',
  Key:'objectkey',
  CopySource:'srcbucketname/srcobjectkey',
  Metadata:{meta1:'value1'}
```

```
},function (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);  
    }  
  }  
});
```

5.6 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
VersionId	String	Optional	Version ID of the object to be deleted

Returned Result (InterfaceResult)

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

Field	Type	Description
DeleteMarker	String	Whether the deleted object is a delete marker
VersionId	String	Version ID of the object to be deleted

Sample Code

```
obsClient.deleteObject({
  Bucket:'bucketname',
  Key:'objectkey'
},function (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);
    }
  }
});
```

5.7 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
Objects	Array	Mandatory	List of objects to be deleted

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

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
DeletedIds		Array	List of successfully deleted objects
-	Key	String	Object name
	VersionId	String	Object version ID
	DeleteMarker	String	Whether the deleted object is a delete marker
	DeleteMarkerVersionId	String	Version ID of the delete marker
Errors		Array	List of objects failed to be deleted
-	Key	String	Object name
	VersionId	String	Object version ID
	Code	String	Error code of the deletion failure
	Message	String	Error message of the deletion failure

Sample Code

```
obsClient.deleteObjects({  
  Bucket:'bucketname',  
  Quiet:false,
```

```
Objects:[{Key:'objectkey'},{Key:'objectkey2'}]
},function (err, result) {
  if(err){
    console.error('Error-->' + err);
  }else{
    if(result.CommonMsg.Status < 300){
      console.log('RequestId-->' + result.InterfaceResult.RequestId);
      console.log('Deleted:');
      for(var 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(var 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);
    }
  }
});
```

5.8 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Field	Type	Optional or Mandatory	Description
VersionId	String	Optional	Object version ID
SseC	String	Optional	Algorithm used in SSE-C decryption. The value can be: • AES256
SseCKey	String	Optional	Key used in SSE-C decryption, which is calculated by using AES-256.

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
LastModified	String	Time when the last modification was made to the object
ContentLength	String	Object size in bytes
ContentType	String	MIME type of the object
ETag	String	Object ETag
VersionId	String	Object version ID
WebsiteRedirectLocation	String	Location where the object is redirected to, when the bucket is configured with website hosting.
StorageClass	String	Storage class of the object. When the storage class is OBS Standard, the value is null.
Restore	String	Restore status of the object in the Archive storage class
Expiration	String	Expiration details
Metadata	Object	Customized metadata of the object You need to add the additional headers allowed to be carried in the response in the CORS configurations. For example, you can add the x-obs-meta-property1 header to obtain customized metadata property1 .

Field	Type	Description
ObjectType	String	Whether the object is an appendable object
NextPosition	String	Start position for next appending

Sample Code

```
obsClient.getObjectMetadata({
  Bucket: 'bucketname',
  Key: 'objectkey'
}, function (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('Code-->' + 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

Field		Type	Optional or Mandatory	Description	
RequestDate		String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.	
VersionId		String	Optional	Object version ID	
ACL		String	Optional	Pre-defined access control policy	
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	Grantee type
		ID	String	Mandatory when Type is Canonical User . In other cases, leave it null.	ID of the domain to which the grantee belongs
		URI	String	Mandatory when Type is Group . In other cases, leave it null.	Grantee group
	Permission		String	Mandatory	Granted permission

 NOTE

- **Owner** and **Grants** must be used together and they cannot be used with **ACL**.
- 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}
  ]
},function (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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
VersionId	String	Optional	Object version ID

Returned Result (InterfaceResult)

Field		Type	Description
RequestId		String	Request ID returned by the OBS server
VersionId		String	Object version ID
Owner		Object	Object owner
-	ID	String	ID of the domain to which the object owner belongs
	Name	String	Name of the object owner
Delivered		String	Whether the bucket ACL is applied to objects in the bucket
Grants		Array	List of grantees' permission information
-	Grantee		Grantee
	-	Name	Grantee name. This field is null when Type of Grantee is Group .
		ID	ID of the domain to which the grantee belongs. This field is null when Type of Grantee is Group .
		URI	Grantee group. This field is null when Type of Grantee is CanonicalUser .
	Permission		Granted permission

Sample Code

```
obsClient.getObjectAcl({
  Bucket:'bucketname',
  Key : 'objectkey'
},function (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(var 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
ACL	String	Optional	Pre-defined access control policy specified during object creation

Field	Type	Optional or Mandatory	Description
StorageClass	String	Optional	Storage class , which can be specified during the object creation
Metadata	Object	Optional	Customized metadata of the object
WebsiteRedirect-Location	String	Optional	Location where the object is redirected to, when the bucket is configured with website hosting.
Expires	Number	Optional	Expiration time of the object, in days
ContentType	String	Optional	MIME type of the object
SseKms	String	Optional	Algorithm used in SSE-KMS encryption. The value can be: kms
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. The value can be null.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: AES256
SseCKey	String	Optional	Key used in SSE-C encryption. It is calculated by using AES-256.

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
Key	String	Name of the object to be uploaded
UploadId	String	Multipart upload ID

Sample Code

```
obsClient.initiateMultipartUpload({  
  Bucket: 'bucketname',  
  Key : 'objectkey',  
})
```

```
    ContentType : 'text/plain'
  },function (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 a 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
PartNumber	Number	Mandatory	Part number, which ranges from 1 to 10000
UploadId	String	Mandatory	Multipart upload ID

Field	Type	Optional or Mandatory	Description
ContentMD5	String	Optional	Base64-encoded MD5 value of the object data to be uploaded. It is provided for the OBS server to verify data integrity.
Body	String	Optional	Object content to be uploaded
SourceFile	File or Blob	Optional	Source file containing the part (The browser must support FileReader .)
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: Number of uploaded bytes, total bytes, and used time (unit: second).
Offset	Number	Optional	Start offset (in bytes) of a part in the source file. The default value is 0 . This field is effective only when SourceFile is set.
PartSize	Number	Optional	Size (in bytes) of a part in the source file. The default value is the file size minus Offset . This field is effective only when SourceFile is set. Except for the part lastly being uploaded whose size ranging from 0 to 5 GB, sizes of the other parts range from 100 KB to 5 GB.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be: AES256
SseCKey	String	Optional	Key used in SSE-C encryption. It is calculated by using AES-256.

 NOTE

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

Returned Result (InterfaceResult)

Field	Type	Description
RequestId	String	Request ID returned by the OBS server
ETag	String	ETag of the uploaded part

Sample Code

```
obsClient.uploadPart({
  Bucket: 'bucketname',
  Key: 'objectkey',
  UploadId: 'uploadid',
  PartNumber: 1,
  Body: 'Hello OBS'
}, function (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 a 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

Field	Type	Optional or Mandatory	Description
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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 range of the source object. The value range is [0, source object length-1] and is in the format of bytes=x-y. If the maximum length of CopySourceRange is larger than the length of the source object minus 1, the length of the source object minus 1 is used.
SseC	String	Optional	Algorithm used to encrypt the target part in SSE-C mode. The value can be: • AES256
SseCKey	String	Optional	Key used to encrypt the target part in SSE-C mode. It is calculated by using AES-256.
CopySourceSseC	String	Optional	Algorithm used to decrypt the source object in SSE-C mode. The value can be: • AES256
CopySourceSseCKey	String	Optional	Key used to decrypt the source object in SSE-C mode, which is calculated by using AES-256

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

Sample Code

```
obsClient.copyPart({
  Bucket:'bucketname',
  Key : 'objectkey',
  PartNumber : 1,
  UploadId : 'uploadid',
  CopySource : 'sourcebucketname/sourceobjectkey',
  CopySourceRange : 'bytes=0-10'
},function (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
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	Mandatory	Bucket name

Field	Type	Optional or Mandatory	Description
Key	String	Mandatory	Object name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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 versioning objects are returned. If the field value is true , not all versioning objects are returned. If the field value is false , all versioning objects 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 initiator belongs
Owner		Object	Owner of the multipart upload, which is consistent with Initiator
-	ID	String	ID of the domain to which the initiator belongs
StorageClass		String	Storage class of the object to be uploaded

Sample Code

```
obsClient.listParts({
  Bucket: 'bucketname',
  Key: 'objectkey',
  UploadId: 'uploadid',
  MaxParts: 10
}, function (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 (var 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
RequestDate		String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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

Field	Type	Description
Key	String	Object name obtained after part combination
Location	String	URL of the object generated after part combination
VersionId	String	Version ID of the object obtained after part combination

Sample Code

```
obsClient.completeMultipartUpload({
  Bucket: 'bucketname',
  Key: 'objectkey',
  UploadId: 'uploadid',
  Parts: [{PartNumber: 1, ETag: 'etag1'}, {PartNumber: 2, ETag: 'etag2'}]
}, function (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 DELETE 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

Field	Type	Optional or Mandatory	Description
Key	String	Mandatory	Object name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
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'
},function (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 Restoring Archive Objects

API Description

You can use this API to restore an Archive object 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
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.
VersionId	String	Optional	Version ID of the to-be-restored Archive object
Days	Number	Mandatory	Retention period of the restored object, in days. The value is from 1 to 30.
Tier	String	Optional	Restore option

Returned Result (InterfaceResult)

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

Sample Code

```
obsClient.restoreObject({
  Bucket: 'bucketname',
  Key: 'objectkey',
  Days: 1,
  Tier: obsClient.enums.RestoreTierExpedited
}, function (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

Table 6-1 Parameter description

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

Field	Type	Optional or Mandatory	Description
SpecialParam	String	Optional	Special operator, which indicates the sub-resource to access. For details, see Table 6-2 .
Expires	Number	Optional	Expiration time of the signed URL, in seconds. The default value is 300 .
Headers	Object	Optional	Headers in the request
QueryParams	Object	Optional	Query parameters in the request

Table 6-2 SubResourceType

Constant	Applicable API
storagePolicy	Specify or obtain bucket storage classes.
quota	Specify or obtain bucket quotas.
storageinfo	Obtain bucket storage information.
location	Obtain bucket locations.
acl	Specify or obtain bucket ACLs or object ACLs.
policy	Specify, obtain, or delete bucket policies.
cors	Specify, obtain, or delete bucket CORS configurations.
versioning	Specify or obtain bucket versioning statuses.
website	Specify, obtain, or delete bucket website configurations.
logging	Specify or obtain bucket logging settings.
lifecycle	Specify, obtain, or delete lifecycle rules of buckets.
notification	Specify or obtain the notification configurations of buckets.
tagging	Specify, obtain, or delete bucket tags.
delete	Batch delete objects.
versions	List object versions in buckets.
uploads	List or initiate multipart uploads in buckets.

Constant	Applicable API
restore	Restore Archive objects.

Returned Result

Table 6-3 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 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-amz-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']));
```

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, which corresponds to the key field in the form
Expires	Number	Optional	Validity period of the browser-based upload authentication, in seconds. The default value is 300 .
FormParams	Object	Optional	Other parameters of the browser-based upload except for key , policy , and signature . Possible values are: • acl • cache-control • content-type • content-disposition • content-encoding • expires

Returned Result

Field	Type	Description
OriginPolicy	String	Value of Policy that is not encoded by base64. This parameter can only be used for verification.
Policy	String	policy in the form
Signature	String	signature in the form

Sample Code

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

6.3 Upload File (Resumable Upload)

API Description

This API is an encapsulated and enhanced version of multipart upload, and supports upload progress reporting, upload event reporting, task pause, and resumable upload.

Method Definition

ObsClient.uploadFile

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	String	If UploadCheckpoint is not set, this parameter is mandatory.	Bucket name
Key	String	If UploadCheckpoint is not set, this parameter is mandatory.	Object name
RequestDate	String or Date	Optional	Request time NOTE When the parameter type is String , the value must comply with the ISO8601 or RFC822 standards.

Field	Type	Optional or Mandatory	Description
SourceFile	File or Blob	If UploadCheckpoint is not set, this parameter is mandatory.	Source file to be uploaded (The browser must support FileReader .)
UploadCheckpoint	Object	Optional	Resumable upload recording object, which can be obtained by running the ResumeCallback command.
PartSize	Number	Optional	Part size, in bytes. The value ranges from 100 KB to 5 GB and defaults to 9 MB.
TaskNum	Number	Optional	Maximum number of parts that can be concurrently uploaded. The default value is 1 .
ProgressCallback	Function	Optional	Callback function for obtaining the upload progress NOTE This callback function contains the following parameters in sequence: Number of uploaded bytes, total bytes, and used time (unit: second).
EventCallback	Function	Optional	Callback function for obtaining the upload event NOTE - The callback function contains the following parameters: event type, event parameter, and event result. - Possible values of the event type are as follows: uploadPartSucceed, uploadPartFailed, uploadPartAborted, initiateMultipartUploadSucceed, initiateMultipartUploadFailed, completeMultipartUploadSucceed, completeMultipartUploadFailed, completeMultipartUploadAborted

Field	Type	Optional or Mandatory	Description
ResumeCallback	Function	Optional	Callback function for obtaining the resumable upload control parameters. NOTE - The callback function contains the following parameters: the control parameter for canceling the resumable upload task and the resumable upload record object. - You can invoke the cancel method of the control parameter for canceling the resumable upload task to suspend a resumable upload task. - The sourceFile field in the resumable upload record object indicates the file to be uploaded. This field needs to be set again after the browser is restarted.
ACL	String	Optional	Pre-defined access control policy specified during object creation
StorageClass	String	Optional	Storage class , which can be specified during the object creation
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 valid value is kms .
SseKmsKey	String	Optional	Master key used in SSE-KMS encryption. The value can be null.
SseC	String	Optional	Algorithm used in SSE-C encryption. The value can be AES256 .

Field	Type	Optional or Mandatory	Description
SseCKey	String	Optional	Key used in SSE-C encryption. It is calculated by using AES-256.

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 generated after part combination
VersionId	String	Version ID of the object obtained after part combination

Sample Code

```
var cp;
var hook;
obsClient.uploadFile({
  Bucket : 'bucketname',
  Key : 'objectkey',
  SourceFile : document.getElementById('input-file').files[0],
  PartSize : 9 * 1024 * 1024,
  ProgressCallback : function(transferredAmount, totalAmount, totalSeconds){
    console.log(transferredAmount * 1.0 / totalSeconds / 1024);
    console.log(transferredAmount * 100.0 / totalAmount);
  },
  ResumeCallback : function(resumeHook, uploadCheckpoint){
    hook = resumeHook;
    cp = uploadCheckpoint;
  }
}, function(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);
    }
  }
}
```

```
}  
});
```

A Change History

Date	What's New
2020-03-31	This is the first official release.