OBS PHP SDK

API Reference

Issue	01
Date	2023-03-14

HUAWEI TECHNOLOGIES CO., LTD.

Security Declaration

Vulnerability

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

Contents

1 Overview

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

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

2 Initialization

2.1 ObsClient Initialization

API Description

ObsClient functions as the PHP client for accessing OBS. It offers callers 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.

Namespace

Class	Parent Namespace
ObsClient	Obs

Method Definition

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

Parameter Description

Field	Type	Optional or Mandatory	Description
key	string	Mandatory	AK
secret	string	Mandatory	SK

Field	Type	Optional or Mandatory	Description
endpoint	string	Mandatory	Endpoint for accessing OBS, which contains the protocol type, domain name (or IP address), and port number. For example, https://your-endpoint:443.
ssl_verify	boolean or string	Optional	Whether to verify server-side certificates. Possible values are: ● Path to the server-side root certificate file in **.pem** format ● **true**: The default CAs are used to verify the server-side certificate. ● **false**: The server-side certificates will not be verified. The default value is **false**.
max_retry_count	integer	Optional	Maximum number of retries when an HTTP/HTTPS connection is abnormal. The default value is **3**.
socket_timeout	integer	Optional	Timeout duration for transmitting data at the socket layer, in seconds. The default value is **60**.
connect_timeout	integer	Optional	Timeout period for establishing an HTTP/HTTPS connection, in seconds. The default value is **60**.
chunk_size	integer	Optional	Block size for reading socket streams, in bytes. The default value is **65536**.

Sample Code

```
// Import the dependency library.
require 'vendor/autoload.php';
// Import the SDK code library during source code installation.
// require 'obs-autoloader.php';
// Declare the namespace.
use Obs\ObsClient;

// Create an instance of ObsClient.
$obsClient = new ObsClient([
      //Obtain an AK/SK pair using environment variables or import the AK/SK pair in other ways. Using hard coding may result in leakage.
      //Obtain an AK/SK pair on the management console. For details, see https://support.huaweicloud.com/eu/usermanual-ca/ca_01_0003.html.
      'key' => getenv('ACCESS_KEY_ID'),
      'secret' => getenv('SECRET_ACCESS_KEY'),
      'endpoint' => 'https://your-endpoint',
      'ssl_verify' => false,
      'max_retry_count' => 1,
      'socket_timeout' => 20,
      'connect_timeout' => 20,
      'chunk_size' => 8196
]);
```

2.2 Log Initialization

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(array $parameter)
```

Parameter Description

Field	Type	Optional or Mandatory	Description
FilePath	string	Mandatory	Save directory of log files
FileName	string	Mandatory	Log file name
MaxFiles	integer	Mandatory	Maximum number of log files that can be retained
Level	integer	Mandatory	Log level. SDK defines four types of integer constant corresponding to different log levels, which are: ● DEBUG (100) ● INFO (200) ● WARN (300) ● ERROR (400)

Sample Code

```
$obsClient->initLog ([
      'FilePath' => './logs',
      'FileName' => 'OBS-SDK.log',
      'MaxFiles' => 10,
      'Level' => INFO
]);
```

2.3 Request Array

API Description

Each time you call an API of **ObsClient**, you need to pass the associative array to the request as the input. For a bucket-related API, the **Bucket** field contained in the associative array is used to specify the bucket name (excluding **ObsClient->listBuckets**). For an object-related API, the **Bucket** field and **Key** field contained in the associative array 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
Other parameters	For details, see "Bucket-Related APIs" and "Object-Related APIs".		

2.4 SDK Common Result Objects

API Description

After an API is called using an instance of **ObsClient**, view whether an exception is thrown. If no, an SDK common result object will be returned, indicating a successful operation. If yes, the operation fails and you need to obtain the error information from the instance of **Obs\Common\ObsException**.

Namespace

Class	Parent Namespace
Model	Obs\Internal\Common

Parameter Description

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Other fields	For details, see "Bucket-Related APIs" and "Object-Related APIs".	

2.5 SDK Custom Exceptions

API Description

SDK custom exceptions are thrown by **ObsClient** and are inherited from class **\RuntimeException**. Exceptions are usually OBS server-side errors, including OBS error codes and error information. This facilitates users to locate problems and troubleshoot faults.

Namespace

Class	Parent Namespace
ObsException	Obs

Method Description

Method	Return Value Type	Description
ObsException->getExceptionCode	string	Error code returned by the OBS server
ObsException->getExceptionMessage	string	Error description returned by the OBS server
ObsException->getRequestId	string	Request ID returned by the OBS server
ObsException->getHostId	string	Requested server ID

Method	Return Value Type	Description
ObsException->getResponse	GuzzleHttp\Psr7\Response	HTTP response object
ObsException->getRequest	GuzzleHttp\Psr7\Request	HTTP request object
ObsException->getStatusCode	integer	HTTP status code

2.6 Asynchronous Method Call

API Description

All bucket- and object-related APIs provided by OBS PHP SDK can be called by asynchronous methods whose names end with **Async** (such as **ObsClient->putObjectAsync** if the synchronous method is named **ObsClient->putObject**). The returned result will be output to a callback function. A callback function contains an **SDK custom exception** and an **SDK common result object** in sequence. If the **SDK custom exception** is not null, the operation fails. Otherwise, the operation succeeds.

Sample Code

```
$promise = $obsClient->putObjectAsync ( [
      'Bucket' => 'bucketname',
      'Key' => 'objectkey',
      'Body' => 'Hello OBS'
], function ($obsException, $resp) {
      if ($obsException === null) {
            printf ( "RequestId:%s\n", $resp ['RequestId'] );
      } else {
            printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
            printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
      }
} );
$promise->wait ();
```

NOTE

A result object (**GuzzleHttp\Promise\Promise**) will be returned upon an asynchronous method call. You need to call the **wait** method of the object to wait until the asynchronous method call is complete.

3 Predefined Constants

3.1 Permission Types

Access Method	Type	Description
ObsClient::PermissionRead	string	A grantee with this permission for a bucket can obtain the list of objects, multipart uploads, and multiple object versions in and the bucket, as well as metadata of the bucket. A grantee with this permission for an object can obtain the object content and metadata.
ObsClient::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::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.

Access Method	Type	Description
ObsClient::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.
ObsClient::PermissionFullCon-trol	string	A grantee with this permission for a bucket has **PermissionRead**, **PermissionWrite**, **PermissionReadAcp**, and **PermissionWriteAcp** permissions for the bucket. A grantee with this permission for an object has **PermissionRead**, **PermissionWriteAcp**, and **PermissionWriteAcp** permissions for the object.

3.2 Available Grantee Groups

Access Method	Type	Description
ObsClient::GranteeGroup	string	Grants permissions to user groups.
ObsClient::GranteeUser	string	Grants permissions to a single user.

3.3 Available Grantee Types

Access Method	Type	Description
ObsClient::GroupAllUsers	string	Indicates all users.
ObsClient::GroupAuthenticatedUs-ers	string	Authorized users. This constant is deprecated.
ObsClient::GroupLogDelivery	string	Log delivery group. This constant is deprecated.

3.4 Pre-defined Access Control Policies

Access Method	Type	Description
ObsClient::AclPrivate	string	Private read/write
ObsClient::AclPublicRead	string	Public read
ObsClient::AclPublicReadWrite	string	Public read/write
ObsClient::AclPublicReadDelivered	string	Public read on a bucket as well as objects in the bucket
ObsClient::AclPublicReadWriteDelivered	string	Public read/write on a bucket as well as objects in the bucket

3.5 Storage Classes

Access Method	Type	Description
ObsClient::StorageClassStandard	string	OBS Standard
ObsClient::StorageClassWarm	string	OBS Infrequent Access
ObsClient::StorageClassCold	string	OBS Archive

3.6 Restore Options

Access Method	Type	Description
ObsClient::RestoreTierExpedited	string	Expedited restoration, which restores an object in 1 to 5 minutes.
ObsClient::RestoreTierStandard	string	Standard restoration, which restores an object in 3 to 5 hours.

3.7 Metadata Replication Policy

Access Method	Type	Description
ObsClient::CopyMetadata	string	Copies metadata.

Access Method	Type	Description
ObsClient::ReplaceMetadata	string	Replaces metadata.

4 Bucket-Related APIs

4.1 PUT Bucket

API Description

You can use this API to create a bucket and name it as you specify. The created bucket name must be unique in OBS. Buckets with the same name can only be created by the same user in the same region. In other cases, creating a bucket with a used name will fail. Each user can create a maximum of 100 buckets.

Method Definition

```
1. ObsClient->createBucket(array $parameter)
2. ObsClient->createBucketAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name A bucket name must comply with the following rules: ● Contains 3 to 63 characters chosen from lowercase letters, digits, hyphens (-), and periods (.), and starts with a digit or letter. ● Cannot be an IP-like address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (.), for example, **my..bucket**. ● Cannot contain periods (.) and hyphens (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
ACL	string	Optional	**Pre-defined access control policy** that can be specified during the bucket creation
StorageClass	string	Optional	Bucket **storage class** that can be specified during the bucket creation
LocationConstraint	string	Mandatory unless the region where the OBS service resides is not the default region.	Region where a bucket will be created. For more information about OBS regions and endpoints, see .

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description

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

Sample Code

```
try{
        $resp = $obsClient -> createBucket([
                'Bucket' => 'bucketname',
                'ACL' => 'private',
                'StorageClass' => ObsClient::StorageClassStandard
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.2 GET Buckets

API Description

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

Method Definition

```
1. ObsClient->listBuckets(array $parameter)
2. ObsClient->listBucketsAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Buckets	indexed array	List of buckets

Field		Type	Description
	Name	string	Bucket name
	CreationDate	string	Time when the bucket is created
	Location	string	Bucket location
Owner		associative array	Bucket owner
	ID	string	ID of the domain to which the bucket owner belongs

Sample Code

```
try{
       $resp = $obsClient -> listBuckets([
              'QueryLocation' => true
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("Owner[ID]:%s\n", $resp['Owner']['ID']);

       foreach ($resp['Buckets'] as $index => $bucket){
              printf("Buckets[%d]\n", $index + 1);
              printf("Name:%s\n", $bucket['Name']);
              printf("CreationDate:%s\n", $bucket['CreationDate']);
              printf("Location:%s\n", $bucket['Location']);
       }
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.3 HEAD Bucket

API Description

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

Method Definition

```
1. ObsClient->headBucket(array $parameter)
2. ObsClient->headBucketAsync(array $parameter, callable callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> headBucket([
                'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("Bucket exists\n");
}catch (Obs\Common\ObsException $obsException){
        if($obsException->getStatusCode() === 404){
                printf("Bucket does not exist\n");
        }else{
                printf("StatusCode:%d\n", $obsException->getStatusCode());
        }
}
```

4.4 DELETE Bucket

API Description

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

Method Definition

```
1. ObsClient->deleteBucket(array $parameter)
2. ObsClient->deleteBucketAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
       $resp = $obsClient -> deleteBucket([
              'Bucket' => 'bucketname'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.5 GET Objects

API Description

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

Method Definition

```
1. ObsClient->listObjects(array $parameter)
2. ObsClient->listObjectsAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Prefix	string	Optional	Prefix that the object names to be listed must contain
Marker	string	Optional	Object name to start with when listing objects in a bucket. All objects are listed in the lexicographical order.
MaxKeys	integer	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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Location	string	Bucket location
Name	string	Bucket name
Delimiter	string	Character used to group object names, which is consistent with that set in the request

Field			Type	Description
IsTruncated			boolean	Whether all objects are returned. If the field value is **true**, not all objects are returned. If the field value is **false**, all objects are returned.
Prefix			string	Object name prefix, which is consistent with that set in the request
Marker			string	Start position for listing objects, which is consistent with that set in the request
NextMarker			string	Object name to start with upon next request for listing objects
MaxKeys			integer	Maximum number of listed objects, which is consistent with that set in the request
Contents			indexed 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		integer	Object size in bytes
	Key		string	Object name
	LastModified		string	Time when the last modification was made to the object
	Owner		associative array	Object owner
		ID	string	ID of the domain to which the object owner belongs
	StorageClass		string	Storage class of the object
CommonPrefixes			indexed 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

```
try{
	$resp = $obsClient -> listObjects([
		'Bucket' => 'bucketname',
		'Prefix' => 'prefix',
		'MaxKeys' => 100
	]);
	printf("RequestId:%s\n", $resp['RequestId']);
	foreach ($resp['Contents'] as $index => $content){
		printf("Contents[%d]\n", $index + 1);
		printf("ETag:%s\n", $content['ETag']);
		printf("Size:%s\n", $content['Size']);
		printf("StorageClass:%s\n", $content['StorageClass']);
		printf("Key:%s\n", $content['Key']);
		printf("LastModified:%s\n", $content['LastModified']);
		printf("Owner[ID]:%s\n", $content['Owner']['ID']);
	}
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.6 GET Object versions

API Description

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

Method Definition

```
1. ObsClient->listVersions(array $parameter)
2. ObsClient->listVersionsAsync(array $parameter, callable $callback)
```

Request Parameter

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

Field	Type	Optional or Mandatory	Description
Delimiter	string	Optional	Character used to group object names. If the object name contains the **Delimiter** parameter, the character string from the first character to the first delimiter 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** does not take effect.

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Location	string	Bucket location
Name	string	Bucket name
Delimiter	string	Character used to group object names, which is consistent with that set in the request
Prefix	string	Object name prefix, which is consistent with that set in the request
IsTruncated	boolean	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.

Field			Type	Description
KeyMarker			string	Start position for listing 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			integer	Maximum number of listed versioning objects, which is consistent with that set in the request
Versions			indexed array	List of versioning objects.
	ETag		string	MD5 value of the object
	Size		integer	Object size in bytes
	Key		string	Object name
	VersionId		string	Object version ID
	IsLatest		boolean	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		associative array	Object owner
		ID	string	ID of the domain to which the object owner belongs
	StorageClass		string	Storage class of the object
DeleteMarkers			indexed array	List of versioning delete markers
	Owner		associative array	Object owner
		ID	string	ID of the domain to which the object owner belongs

Field		Type	Description
	Key	string	Object name
	VersionId	string	Object version ID
	IsLatest	boolean	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		indexed 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

```
try{
       $resp = $obsClient -> listVersions([
              'Bucket' => 'bucketname',
              'Prefix' => 'prefix',
              'MaxKeys' => 100
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("Versions:\n");
       foreach ($resp['Versions'] as $index => $version){
              printf("Versions[%d]\n", $index + 1);
              printf("ETag:%s\n", $version['ETag']);
              printf("Size:%s\n", $version['Size']);
              printf("StorageClass:%s\n", $version['StorageClass']);
              printf("Key:%s\n", $version['Key']);
              printf("VersionId:%s\n", $version['VersionId']);
              printf("LastModified:%s\n", $version['LastModified']);
              printf("Owner[ID]:%s\n", $version['Owner']['ID']);
       }

       printf("DeleteMarkers:\n");
       foreach ($resp['DeleteMarkers'] as $index => $deleteMarker){
              printf("DeleteMarkers[%d]\n", $index + 1);
              printf("Key:%s\n", $deleteMarker['Key']);
              printf("VersionId:%s\n", $deleteMarker['VersionId']);
              printf("IsLatest:%s\n", $deleteMarker['IsLatest']);
              printf("LastModified:%s\n", $deleteMarker['LastModified']);
              printf("Owner[ID]:%s\n", $deleteMarker['Owner']['ID']);
        }
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.7 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

```
1. ObsClient->listMultipartUploads(array $parameter)
2. ObsClient->listMultipartUploadsAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Delimiter	string	Optional	Character used to group object names involved in multipart uploads. If the object name contains the **Delimiter** parameter, the character string from the first character to the first delimiter 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	integer	Optional	Maximum number of returned multipart uploads. The value ranges from 1 to 1000. If the value is not in this range, 1000 is returned by default.
KeyMarker	string	Optional	Object name to start with when listing multipart uploads
UploadIdMarker	string	Optional	Upload ID after which the multipart upload listing begins. It is effective only when used with **KeyMarker** so that multipart uploads after **UploadIdMarker** of **KeyMarker** will be listed.

Returned Result

Field		Type	Description
HttpStatusCode		integer	HTTP status code
Reason		string	Reason 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		integer	Maximum number of listed multipart uploads, which is consistent with the same parameter in the request
IsTruncated		boolean	Whether all multipart uploads are returned. If the field value is **true**, not all multipart uploads are returned. If the field value is **false**, all multipart uploads are returned.
Uploads		indexed array	List of multipart uploads.
	Key	string	Name of the object to be uploaded
	UploadId	string	Multipart upload ID
	Initiator	associative array	Initiator of the multipart upload

Field			Type	Description
		ID	string	ID of the domain to which the initiator belongs
	Owner		associative array	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			indexed 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

```
try{
       $resp = $obsClient -> listMultipartUploads([
              'Bucket' => 'bucketname',
              'Prefix' => 'prefix',
              'MaxUploads' => 100
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       foreach ($resp['Uploads'] as $index => $upload){
              printf("Versions[%d]\n", $index + 1);
              printf("UploadId:%s\n", $upload['UploadId']);
              printf("Initiated:%s\n", $upload['Initiated']);
              printf("StorageClass:%s\n", $upload['StorageClass']);
              printf("Key:%s\n", $upload['Key']);
              printf("Initiator[ID]:%s\n", $upload['Initiator']['ID']);
              printf("Initiator[DisplayName]:%s\n", $upload['Initiator']['DisplayName']);
              printf("Owner[ID]:%s\n", $upload['Owner']['ID']);
              printf("Owner[DisplayName]:%s\n", $upload['Owner']['DisplayName']);
       }
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.8 Obtain Bucket Metadata

API Description

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

Method Definition

```
1. ObsClient->getBucketMetadata(array $parameter)
2. ObsClient->getBucketMetadata(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Location	string	Bucket location
StorageClass	string	Storage class of the bucket. When the storage class is OBS Standard, the value is null.
AllowOrigin	string	If **Origin** in the request meets the CORS rules of the server, **AllowedOrigin** in the CORS rules is returned.
AllowHeader	string	If **RequestHeader** in the request meets the CORS rules of the server, **AllowedHeader** in the CORS rules is returned.
AllowMethod	string	**AllowedMethod** in the CORS rules of the server
ExposeHeader	string	**ExposeHeader** in the CORS rules of the server

Field	Type	Description
MaxAgeSeconds	integer	**MaxAgeSeconds** in the CORS rules of the server

Sample Code

```
try{
        $resp = $obsClient -> getBucketMetadata([
                'Bucket' => 'bucketname',
                'Origin' => 'http://www.a.com',
                'RequestHeader' => 'x-obs-header'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("StorageClass:%s\n", $resp['StorageClass']);
        printf("AllowOrigin:%s\n", $resp['AllowOrigin']);
        printf("AllowHeader:%s\n", $resp['AllowHeader']);
        printf("AllowMethod:%s\n", $resp['AllowMethod']);
        printf("ExposeHeader:%s\n", $resp['ExposeHeader']);
        printf("MaxAgeSeconds:%s\n", $resp['MaxAgeSeconds']);
}catch (Obs\Common\ObsException $obsException){
        printf("StatusCode:%s\n", $obsException->getStatusCode());
}
```

4.9 GET Bucket location

API Description

You can use this API to obtain the bucket location.

Method Definition

```
1. ObsClient->getBucketLocation(array $parameter)
2. ObsClient->getBucketLocationAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description

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

Sample Code

```
try{
        $resp = $obsClient -> getBucketLocation([
                'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("Location:%s\n", $resp['Location']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.10 GET Bucket storageinfo

API Description

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

Method Definition

```
1. ObsClient->getBucketStorageInfo(array $parameter)
2. ObsClient->getBucketStorageInfo(array $parameter, callback $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result (InterfaceResult)

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Size	double	Bucket size

Field	Type	Description
ObjectNumber	integer	Number of objects in the bucket

Sample Code

```
try{
        $resp = $obsClient -> getBucketStorageInfo([
                'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("Size:%s\n", $resp['Size']);
        printf("ObjectNumber:%s\n", $resp['ObjectNumber']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.11 PUT Bucket quota

API Description

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

Method Definition

```
1. ObsClient->setBucketQuota(array $parameter)
2. ObsClient->setBucketQuotaAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description

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

Sample Code

```
try{
       $resp = $obsClient -> setBucketQuota([
              'Bucket' => 'bucketname',
              'StorageQuota' => 100 * 1024 * 1024
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.12 GET Bucket quota

API Description

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

Method Definition

```
1. ObsClient->getBucketQuota(array $parameter)
2. ObsClient->getBucketQuotaAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
StorageQuota	integer	Bucket quota

Sample Code

```
try{
      $resp = $obsClient -> getBucketQuota([
            'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
      printf("StorageQuota:%s\n", $resp['StorageQuota']);
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.13 Set Bucket storagePolicy

API Description

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

Method Definition

```
1. ObsClient->setBucketStoragePolicy(array $parameter)
2. ObsClient->setBucketStoragePolicyAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
      $resp = $obsClient -> setBucketStoragePolicy([
            'Bucket' => 'bucketname',
            'StorageClass' => ObsClient::StorageClassWarm
```

```
	]);
	printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.14 GET Bucket storagePolicy

API Description

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

Method Definition

```
1. ObsClient->getBucketStoragePolicy(array $parameter)
2. ObsClient->getBucketStoragePolicyAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
StorageClass	string	Storage class of the bucket

Sample Code

```
try{
	$resp = $obsClient -> getBucketStoragePolicy([
		'Bucket' => 'bucketname'
	]);
	printf("RequestId:%s\n", $resp['RequestId']);
	printf("StorageClass:%s\n", $resp['StorageClass']);
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.15 PUT Bucket acl

API Description

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

Method Definition

```
1. ObsClient->setBucketAcl(array $parameter)
2. ObsClient->setBucketAclAsync(array $parameter, callable $callback)
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			string	Mandatory	Bucket name
ACL			string	Optional	**Pre-defined access control policy**
Owner			associative array	Optional	Bucket owner
	ID		string	Mandatory	ID of the domain to which the bucket owner belongs
	DisplayName		string	Optional	Name of the bucket owner
Grants			indexed array	Optional	List of grantees' permission information
	Grantee		associative array	Mandatory	Grantee
		Type	string	Mandatory	**Grantee type**

Field			Type	Optional or Mandatory	Description
		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**
	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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> setBucketAcl([
                'Bucket' => 'bucketname',
                'Owner' => ['ID' => 'ownerid', 'DisplayName' => 'ownername'],
                'Grants' => [
                        ['Grantee' => ['Type' => 'CanonicalUser', 'ID' => 'userid'], 'Permission' =>
ObsClient::PermissionRead],
                        ['Grantee' => ['Type' => 'CanonicalUser', 'ID' => 'userid'], 'Permission' =>
ObsClient::PermissionWrite],
                        ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupLogDelivery], 'Permission' =>
ObsClient::PermissionWrite],
                        ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupLogDelivery], 'Permission' =>
ObsClient::PermissionReadAcp],
                ]
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.16 GET Bucket acl

API Description

You can use this API to obtain a bucket ACL.

Method Definition

```
1. ObsClient->getBucketAcl(array $parameter)
2. ObsClient->getBucketAclAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Field			Type	Description
Owner			associative array	Bucket owner
	ID		string	ID of the domain to which the bucket owner belongs
Grants			indexed array	List of grantees' permission information
	Grantee		associative array	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
	Delivered		boolean	Whether an object inherits the ACL of its residing bucket

Sample Code

```
try{
	$resp = $obsClient -> getBucketAcl([
		'Bucket' => 'bucketname'
	]);
	printf("RequestId:%s\n", $resp['RequestId']);
	printf("Owner[ID]:%s\n", $resp['Owner']['ID']);
	printf("Owner[DisplayName]:%s\n", $resp['Owner']['DisplayName']);
	printf("Grants\n");
	foreach ($resp['Grants'] as $index => $grant){
		printf("Grants[%d]", $index + 1);
		printf("Grantee[ID]:%s\n", $grant['Grantee']['ID']);
		printf("Grantee[DisplayName]:%s\n", $grant['Grantee']['DisplayName']);
		printf("Grantee[URI]:%s\n", $grant['Grantee']['URI']);
		printf("Permission:%s\n", $grant['Permission']);
	}
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.17 PUT Bucket logging

API Description

You can use this API to configure access logging for a bucket.

Method Definition

```
1. ObsClient->setBucketLogging(array $parameter)
2. ObsClient->setBucketLoggingAsync(array $parameter, callable $callback)
```

Request Parameter

Field				Type	Optional or Mandatory	Description
Bucket				string	Mandatory	Bucket name
Agency				string	Mandatory when configuring bucket logging	Agency name
LoggingEnabled				associative array	Optional	Log configuration information
	TargetBucket			string	Mandatory	Target bucket for which logs are generated
	TargetPrefix			string	Mandatory	Name prefix of a to-be-logged object in the target bucket
	TargetGrants			indexed array	Optional	List of grantees' permission information
		Grantee		associative array	Optional	Grantee
			Type	string	Mandatory	**Grantee type**

Field				Type	Optional or Mandatory	Description
			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**
		Permission		string	Optional	**Granted permission**

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
       $resp = $obsClient -> setBucketLogging([
              'Bucket' => 'bucketname',
              'LoggingEnabled' => [
                     'TargetBucket' => 'targetbucketname',
                     'TargetPrefix' => 'prefix',
                     'TargetGrants' => [
                           ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupAuthenticatedUsers],
'Permission' => ObsClient::PermissionWriteAcp],
                           ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupAuthenticatedUsers],
'Permission' => ObsClient::PermissionRead]
                     ]
              ]
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.18 GET Bucket logging

API Description

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

Method Definition

```
1. ObsClient->getBucketLogging(array $parameter)
2. ObsClient->getBucketLoggingAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field				Type	Description
HttpStatusCode				integer	HTTP status code
Reason				string	Reason description
RequestId				string	Request ID returned by the OBS server
Agency				string	Agency name
LoggingEnabled				associative array	Log configuration information
	TargetBucket			string	Target bucket for which logs are generated
	TargetPrefix			string	Name prefix of a to-be-logged object in the target bucket
	TargetGrants			indexed array	List of grantees' permission information
		Grantee		associative array	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

```
try{
      $resp = $obsClient -> getBucketLogging([
             'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
      printf("LoggingEnabled[TargetBucket]:%s\n", $resp['LoggingEnabled']['TargetBucket']);
      printf("LoggingEnabled[TargetPrefix]:%s\n", $resp['LoggingEnabled']['TargetPrefix']);
      printf("TargetGrants\n");
      foreach ($resp['LoggingEnabled']['TargetGrants'] as $index => $grant){
             printf("Grants[%d]", $index + 1);
             printf("Grantee[ID]:%s\n", $grant['Grantee']['ID']);
             printf("Grantee[DisplayName]:%s\n", $grant['Grantee']['DisplayName']);
             printf("Grantee[URI]:%s\n", $grant['Grantee']['URI']);
             printf("Permission:%s\n", $grant['Permission']);
      }
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.19 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

```
1. ObsClient->setBucketPolicy(array $parameter)
2. ObsClient->setBucketPolicyAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> setBucketPolicy([
                'Bucket' => 'bucketname',
                'Policy' => 'your policy'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

NOTICE

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

4.20 GET Bucket policy

API Description

You can use this API to obtain the bucket policy.

Method Definition

```
1. ObsClient->getBucketPolicy(array $parameter)
2. ObsClient->getBucketPolicyAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Policy	string	Policy information in JSON format

Sample Code

```
try{
       $resp = $obsClient -> getBucketPolicy([
              'Bucket' => 'bucketname'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("Policy:%s\n", $resp['Policy']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.21 DELETE Bucket policy

API Description

You can use this API to delete a bucket policy.

Method Definition

```
1. ObsClient->deleteBucketPolicy(array $parameter)
2. ObsClient->deleteBucketPolicyAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
      $resp = $obsClient -> deleteBucketPolicy([
            'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.22 PUT Bucket lifecycle

API Description

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

Method Definition

```
1. ObsClient->setBucketLifecycle(array $parameter)
2. ObsClient->setBucketLifecycleAsync(array $parameter, callable $callback)
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			string	Mandatory	Bucket name
Rules			indexed array	Mandatory	Lifecycle rules of the bucket
	Transitions		indexed array	Optional	List of object transition policies
		StorageClass	string	Mandatory	**Storage class** of the object after transition **NOTE** The Standard storage class is not supported.
		Date	string or \DateTime	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.

Field			Type	Optional or Mandatory	Description
		Days	integer	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		associative array	Optional	Expiration time of an object
		Date	string or \DateTime	Mandatory when **Days** is not set	Date when an object expires. If the value type is **string**, the value must conform to the ISO8601 standards and must be at 00:00 (UTC time), for example, 2018-01-01T00:00:00Z.
		Days	integer	Mandatory when **Date** is not set	Number of days after which an object expires since its creation. The value must be a positive integer.

Field			Type	Optional or Mandatory	Description
	ID		string	Optional	Rule ID. It is a 1-255 character string.
	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		indexed array	Optional	List of noncurrent object version transition policies
		StorageClass	string	Mandatory	**Storage class** of the noncurrent object version after transition

Field			Type	Optional or Mandatory	Description
		NoncurrentDays	integer	Mandatory	Number of days after which an object will be transited since it becomes a noncurrent version. The parameter value must be a positive integer.
	NoncurrentVersionExpiration		associative array	Optional	Expiration time of a noncurrent object version
		NoncurrentDays	integer	Mandatory	Number of days after which an object expires since it becomes a noncurrent version. The field value must be a positive integer.

NOTE

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
       $resp = $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']]
              ]
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.23 GET Bucket lifecycle

API Description

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

Method Definition

```
1. ObsClient->getBucketLifecycle(array $parameter)
2. ObsClient->getBucketLifecycleAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field			Type	Description
HttpStatusCode			integer	HTTP status code
Reason			string	Reason description
RequestId			string	Request ID returned by the OBS server
Rules			indexed array	Lifecycle rules of the bucket
	Transitions		indexed 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		associative array	Expiration time of an object
		Date	string	Date when an object expires
		Days	integer	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		indexed array	List of noncurrent object version transition policies
		Storage Class	string	Storage class of the noncurrent object version after transition
		NoncurrentDays	string	Number of days after which an object will be transited since it becomes a noncurrent version
	NoncurrentVersionExpiration		associative array	Expiration time of a noncurrent object version
		NoncurrentDays	integer	Number of days after which an object expires since it becomes a noncurrent version

Sample Code

```
try{
        $resp = $obsClient -> getBucketLifecycle([
                'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        foreach ($resp['Rules'] as $index => $rule){
                printf("Rules[%d]\n", $index + 1);
                printf("ID:%s\n", $rule['ID']);
                printf("Prefix:%s\n", $rule['Prefix']);
                printf("Status:%s\n", $rule['Status']);
                printf("Expiration[Days]:%s\n", $rule['Expiration']['Days']);
                printf("Expiration[Date]:%s\n", $rule['Expiration']['Date']);
                printf("NoncurrentVersionExpiration[NoncurrentDays]:%s\n", $rule['NoncurrentVersionExpiration']
['NoncurrentDays']);
        }
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.24 DELETE Bucket lifecycle

API Description

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

Method Definition

```
1. ObsClient->deleteBucketLifecycle(array $parameter)
2. ObsClient->deleteBucketLifecycleAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
      $resp = $obsClient -> deleteBucketLifecycle([
            'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.25 PUT Bucket website

API Description

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

Method Definition

```
1. ObsClient->setBucketWebsite(array $parameter)
2. ObsClient->setBucketWebsiteAsync(array $parameter, callable $callback)
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		string	Mandatory	Bucket name
RedirectAllRequestsTo		associative array	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		associative array	Optional	Error page settings
	Key	string	Optional	Page that is returned when a **4*XX*** error occurs
IndexDocument		associative array	Optional	Default page settings

Field			Type	Optional or Mandatory	Description
	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			indexed array	Optional	Redirection rule list
	Condition		associative array	Optional	Matching condition of a redirection rule
		HttpErrorCodeReturnedEquals	string	Optional	HTTP error code to be matched when a redirection rule takes effect
		KeyPrefixEquals	string	Optional	Object name prefix to be matched when a redirection rule takes effect
	Redirect		associative array	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 **ReplaceKeyPrefix-With**.
		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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> setBucketWebsite([
                'Bucket' => 'bucketname',
//              'RedirectAllRequestsTo' => ['HostName' => 'www.example.com', 'Protocol' => 'https'],
                'IndexDocument' => ['Suffix' => 'index.html'],
                'ErrorDocument' => ['Key' => 'error.html'],
                'RoutingRules' => [
                        ['Condition' => ['HttpErrorCodeReturnedEquals' => 404, 'KeyPrefixEquals' => 'prefix'],
'Redirect' => ['Protocol' => 'http', 'ReplaceKeyWith' => 'key']],
                        ['Condition' => ['HttpErrorCodeReturnedEquals' => 404, 'KeyPrefixEquals' => 'prefix'],
'Redirect' => ['Protocol' => 'http', 'ReplaceKeyWith' => 'key']]
                ]
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.26 GET Bucket website

API Description

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

Method Definition

```
1. ObsClient->getBucketWebsite(array $parameter)
2. ObsClient->getBucketWebsiteAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field			Type	Description
HttpStatusCode			integer	HTTP status code
Reason			string	Reason description
RequestId			string	Request ID returned by the OBS server
RedirectAllRequestsTo			associative array	Redirection rule of all requests
	HostName		string	Host name used for redirection
	Protocol		string	Host name used for redirection
ErrorDocument			associative array	Error page settings
	Key		string	Page that is returned when a **4***XX* error occurs
IndexDocument			associative array	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 (/).

Field			Type	Description
RoutingRules			indexed array	Redirection rule list
	Condition		associative array	Matching condition of a redirection rule
		HttpErrorCodeReturnedEquals	integer	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		associative array	Details about a redirection request
		Protocol	string	Protocol used for redirection
		HostName	string	Host name used for redirection
		ReplaceKeyPrefixWith	string	Object name prefix used in the redirection request
		ReplaceKeyWith	string	Object name used in the redirection request. This parameter cannot be used together with **ReplaceKeyPrefixWith**.
		HttpRedirectCode	integer	HTTP status code in the response to the redirection request

Sample Code

```
try{
       $resp = $obsClient -> getBucketWebsite([
              'Bucket' => 'bucketname'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("ErrorDocument[Key]:%s\n", $resp['ErrorDocument']['Key']);
       printf("IndexDocument[Suffix]:%s\n", $resp['IndexDocument']['Suffix']);
       foreach ($resp['RoutingRules'] as $index => $routingRule){
              printf("RoutingRules[%d]", $index + 1);
              printf("Condition[HttpErrorCodeReturnedEquals]:%s\n", $routingRule['Condition']
['HttpErrorCodeReturnedEquals']);
              printf("Condition[KeyPrefixEquals]:%s\n", $routingRule['Condition']['KeyPrefixEquals']);
              printf("Redirect[Protocol]:%s\n", $routingRule['Redirect']['Protocol']);
              printf("Redirect[ReplaceKeyWith]:%s\n", $routingRule['Redirect']['ReplaceKeyWith']);
              printf("Redirect[HttpRedirectCode]:%s\n", $routingRule['Redirect']['HttpRedirectCode']);
              printf("Redirect[HostName]:%s\n", $routingRule['Redirect']['HostName']);
       }
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.27 DELETE Bucket website

API Description

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

Method Definition

```
1. ObsClient->deleteBucketWebsite(array $parameter)
2. ObsClient->deleteBucketWebsiteAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> deleteBucketWebsite([
               'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.28 PUT Bucket versioning

API Description

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

Method Definition

```
1. ObsClient->setBucketVersioning(array $parameter)
2. ObsClient->getBucketVersioningAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Status	string	Mandatory	Versioning status of the bucket. Possible values are: ● Enabled ● Suspended

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
       $resp = $obsClient -> setBucketVersioning([
              'Bucket' => 'bucketname',
              'Status' => 'Enabled'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.29 GET Bucket versioning

API Description

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

Method Definition

```
1. ObsClient->getBucketVersioning(array $parameter)
2. ObsClient->getBucketVersioningAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Status	string	Versioning status of the bucket

Sample Code

```
try{
        $resp = $obsClient -> getBucketVersioning([
                'Bucket' => 'bucketname'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("Status:%s\n", $resp['Status']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.30 PUT Bucket cors

API Description

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

Method Definition

```
1. ObsClient->setBucketCors(array $parameter)
2. ObsClient->setBucketCorsAsync(array $parameter, callable $callback)
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		string	Mandatory	Bucket name
CorsRules		indexed array	Mandatory	CORS rules of the bucket
	ID	string	Optional	CORS rule ID. It is a 1-255 character string.
	AllowedMethod	indexed array of strings	Mandatory	HTTP methods allowed by the CORS rule. Possible values are: ● GET ● PUT ● HEAD ● POST ● DELETE
	AllowedOrigin	indexed array of strings	Mandatory	Origins (character strings representing domain names) allowed by the CORS rule. Each **AllowedOrigin** can contain up to one wildcard character (*).
	AllowedHeader	indexed array of strings	Optional	Request headers allowed by the CORS rule. Each **AllowedHeader** can contain up to one wildcard character (*).
	MaxAgeSeconds	integer	Optional	Cache duration (in seconds) of the cross-region request result in the client allowed by the CORS rule. The value must be an integer.
	ExposeHeader	indexed array of strings	Optional	Additional response headers allowed by the CORS rule. It cannot contain spaces.

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description

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

Sample Code

```
try{
        $resp = $obsClient -> setBucketCors([
                'Bucket' => 'bucketname',
                'CorsRules' => [
                        [
                              'ID' => 'rule1',
                              'AllowedMethod' => ['PUT','POST','GET','DELETE','HEAD'],
                              'AllowedOrigin' => ['obs.hostname','obs.hostname1'],
                              'AllowedHeader' => ['obs-header-1'],
                              'MaxAgeSeconds' => 60
                        ],
                        [
                                      'ID' => 'rule2',
                                      'AllowedMethod' => ['PUT','POST','GET'],
                                      'AllowedOrigin' => ['obs.hostname','obs.hostname1'],
                                      'AllowedHeader' => ['header-1','header-2'],
                                      'MaxAgeSeconds' => 50
                        ]
                ]
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
        printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.31 GET Bucket cors

API Description

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

Method Definition

```
1. ObsClient->setBucketCors(array $parameter)
2. ObsClient->setBucketCorsAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field		Type	Description
HttpStatusCode		integer	HTTP status code
Reason		string	Reason description
RequestId		string	Request ID returned by the OBS server
CorsRules		indexed array	CORS rules of the bucket
	ID	string	CORS rule ID
	AllowedMethod	indexed array of strings	HTTP methods allowed by the CORS rule
	AllowedOrigin	indexed array of strings	Origins (character strings representing domain names) allowed by the CORS rule
	AllowedHeader	indexed array of strings	Request headers allowed by the CORS rule
	MaxAgeSecond	integer	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	indexed array of strings	Additional response headers allowed by the CORS rule

Sample Code

```
try{
	$resp = $obsClient -> getBucketCors([
		'Bucket' => 'bucketname'
	]);
	printf("RequestId:%s\n", $resp['RequestId']);
	foreach ($resp['CorsRules'] as $index => $corsRule){
		printf("CorsRule[%d]\n", $index + 1);
		printf("MaxAgeSeconds:%s\n", print_r($corsRule['MaxAgeSeconds'], true));
		printf("AllowedMethod:%s\n", print_r($corsRule['AllowedMethod'], true));
		printf("AllowedOrigin:%s\n", print_r($corsRule['AllowedOrigin'], true));
		printf("AllowedHeader:%s\n", print_r($corsRule['AllowedHeader'], true));
		printf("ExposeHeader:%s\n", print_r($corsRule['ExposeHeader'],true));
	}
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.32 DELETE Bucket cors

API Description

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

Method Definition

```
1. ObsClient->setBucketCors(array $parameter)
2. ObsClient->setBucketCorsAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
      $resp = $obsClient -> deleteBucketCors([
            'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.33 PUT Bucket tagging

API Description

You can use this API to set bucket tags.

Method Definition

```
1. ObsClient->setBucketTagging(array $parameter)
2. ObsClient->setBucketTaggingAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
        $resp = $obsClient -> setBucketTagging([
                'Bucket' => 'bucketname',
                'Tags' => [
                        ['Key' => 'tag1', 'Value' => 'value1'],
                        ['Key' => 'tag2', 'Value' => 'value2']
                ]
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
        printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
```

```
    printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.34 GET Bucket tagging

API Description

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

Method Definition

```
1. ObsClient->getBucketTagging(array $parameter)
2. ObsClient->getBucketTaggingAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field		Type	Description
HttpStatusCode		integer	HTTP status code
Reason		string	Reason description
RequestId		string	Request ID returned by the OBS server
Tags		indexed 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

```
try{
    $resp = $obsClient -> getBucketTagging([
        'Bucket' => 'bucketname'
    ]);
    printf("RequestId:%s\n", $resp['RequestId']);
    foreach ($resp['Tags'] as $index => $tag){
            printf("TagSet[%d]\n", $index + 1);
            printf("Key:%s\n", $tag['Key']);
            printf("Value:%s\n", $tag['Value']);
      }
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

4.35 DELETE Bucket tagging

API Description

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

Method Definition

```
1. ObsClient->deleteBucketTagging(array $parameter)
2. ObsClient->deleteBucketTaggingAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try{
      $resp = $obsClient -> deleteBucketTagging([
                  'Bucket' => 'bucketname'
      ]);
      printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
      printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
      printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

5 Objects-Related APIs

5.1 PUT Object

API Description

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

Method Definition

```
1. ObsClient->putObject(array $parameter)
2. ObsClient->putObjectAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
ACL	string	Optional	**Pre-defined access control policy** specified during object creation
StorageClass	string	Optional	**Storage class**, which can be specified during the object creation

Field	Type	Optional or Mandatory	Description
Body	string or resource or GuzzleHttp\Psr7\StreamInterface	Optional	Object content to be uploaded
SourceFile	string	Optional	Path to the source file of the object
Metadata	associative array	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.
ContentType	string	Optional	MIME type of the object
ContentLength	integer	Optional	Object size in bytes
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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
ObjectURL	string	Object URL
ETag	string	Object ETag
VersionId	string	Object version ID
StorageClass	string	Storage class of the object. When the storage class is OBS Standard, the value is null.
SseKms	string	Algorithm used in SSE-KMS encryption
SseKmsKey	string	Key used in SSE-KMS encryption
SseC	string	Algorithm used in SSE-C encryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C encryption

Sample Code

```
try{
       $resp = $obsClient -> putObject([
              'Bucket' => 'bucketname',
              'Key' => 'objectkey',
              'Metadata' => ['meta1' => 'value1', 'meta2' => 'value2'],
//            'SourceFile' => 'localfile',
              'Body' => 'Hello OBS',
              'ContentType' => 'text/plain'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("VersionId:%s\n", $resp['VersionId']);
       printf("StorageClass:%s\n", $resp['StorageClass']);
       printf("ETag:%s\n", $resp['ETag']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

5.2 GET Object

API Description

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

Method Definition

```
1. ObsClient->getObject(array $parameter)
2. ObsClient->getObjectAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
VersionId	string	Optional	Object version ID
IfMatch	string	Optional	Returns the source object if its ETag is the same as the one specified by this parameter; otherwise, an exception is thrown.
IfModifiedSince	string or \DateTime	Optional	Returns the object if it is modified after the time specified by this parameter; otherwise, an exception is thrown. If this parameter value is a character string, it 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 exception is thrown.
IfUnmodifiedSince	string or \DateTime	Optional	Returns the object if it remains unchanged since the time specified by this parameter; otherwise, an exception is thrown. If this parameter value is a character string, it 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.
Origin	string	Optional	Origin of the cross-domain request specified by the pre-request. Generally, it is a domain name.

Field	Type	Optional or Mandatory	Description
RequestHeader	string	Optional	HTTP 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.
SaveAsFile	string	Optional	Target path to which the object is downloaded (containing the file name)
SaveAsStream	boolean	Optional	Whether to return the object in the format of data stream
FilePath	string	Optional	Target path to which the object is downloaded (containing the file name). This is a deprecated parameter and is used to maintain compatibility with earlier versions.
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 **SaveAsStream** is **true**, it cannot be used with **SaveAsFile** or **FilePath**.
- **SaveAsFile** and **FilePath** cannot be used together.
- 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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
DeleteMarker	boolean	Whether the deleted object is a delete marker
LastModified	string	Time when the last modification was made to the object
ContentLength	integer	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 **STANDARD**, the value is null.
Restore	string	Restore status of the object in the OBS Archive storage class

Field	Type	Description
AllowOrigin	string	If **Origin** in the request meets the CORS rules of the bucket, **AllowedOrigin** in the CORS rules is returned.
AllowHeader	string	If **RequestHeader** in the request meets the CORS rules of the bucket, **AllowedHeader** in the CORS rules is returned.
AllowMethod	string	**AllowedMethod** in the CORS rules of the bucket
ExposeHeader	string	**ExposeHeader** in the CORS rules of the bucket
MaxAgeSeconds	string	**MaxAgeSeconds** in the CORS rules of the bucket
SseKms	string	Algorithm used in SSE-KMS decryption
SseKmsKey	string	Master key used in SSE-KMS decryption
SseC	string	Algorithm used in SSE-C decryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C decryption
Expiration	string	Expiration details
Body	GuzzleHttp \Psr7\Stream	Object content. If **SaveAsFile** is set, this field is null. If **SaveAsStream** is set to **true**, this field is a readable stream. You need to call the **GuzzleHttp \Psr7\Stream->read** method to read the data.
SaveAsFile	string	Target path to which the object is downloaded (containing the file name), which is consistent with that set in the request
Metadata	associative array	Customized metadata of the object

Sample Code

```
try{
        $resp = $obsClient -> getObject([
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
//              'SaveAsFile' => 'localfile',
//              'SaveAsStream' => true,
                'Range' => 'bytes=0-10'
        ]);
        printf("RequestId:%s\n", $resp['RequestId']);
        printf("ETag:%s\n", $resp['ETag']);
```

```
	printf("VersionId:%s\n", $resp['VersionId']);
	printf("StorageClass:%s\n", $resp['StorageClass']);
	printf("ContentLength:%s\n", $resp['ContentLength']);
	printf("DeleteMarker:%s\n", $resp['DeleteMarker']);
	printf("LastModified:%s\n", $resp['LastModified']);
	printf("Body:%s\n", $resp['Body']);
	printf("Metadata:%s\n", print_r($resp['Metadata'], true));
}catch (Obs\Common\ObsException $obsException){
	printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
	printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

5.3 PUT Object - Copy

API Description

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

Method Definition

```
1. ObsClient->copyObject(array $parameter)
2. ObsClient->copyObjectAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Target bucket name
Key	string	Mandatory	Target object name
ACL	string	Optional	**Pre-defined access control policy** specified during object copy
StorageClass	string	Optional	**Storage class** of the object. Possible values are:
CopySource	string	Mandatory	Parameter used to specify the source bucket, source object, and source object version ID which can be null. It is in the format of *SourceBucketName/SourceObjectName?versionId=SourceObjectVersionId*.
CopySourceIfMatch	string	Optional	Copies the source object if its ETag is the same as the one specified by this parameter; otherwise, an exception is thrown.

Field	Type	Optional or Mandatory	Description
CopySourceIfModifiedSince	string or \DateTime	Optional	Copies the source object if it is changed after the time specified by this parameter; otherwise, an exception is thrown. If this parameter value is a character string, it 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 exception is thrown.
CopySourceIfUnmodifiedSince	string or \DateTime	Optional	Copies the source object if it is changed before the time specified by this parameter; otherwise, an exception is thrown. If this parameter value is a character string, it 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	associative array	Optional	Customized metadata of the target object
WebsiteRedirect-Location	string	Optional	Location where the object is redirected to, when the bucket is configured with website hosting.

Field	Type	Optional or Mandatory	Description
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
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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
ETag	string	ETag of the target object

Field	Type	Description
LastModified	string	Time when the last modification was made to the object
VersionId	string	Version ID of the target object. This field is null if versioning is not enabled for the target bucket.
CopySourceVersionId	string	Version ID of the source object. This field is null if versioning is not enabled for the source bucket.
SseKms	string	Algorithm used in SSE-KMS encryption
SseKmsKey	string	Master key used in SSE-KMS encryption
SseC	string	Algorithm used in SSE-C encryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C encryption

Sample Code

```
try{
       $resp = $obsClient -> copyObject([
              'Bucket' => 'bucketname',
              'Key' => 'objectkey',
              'CopySource' => 'srcbucketname/srcobjectkey',
              'Metadata' => ['meta1' => 'value1']
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
       printf("ETag:%s\n", $resp['ETag']);
       printf("VersionId:%s\n", $resp['VersionId']);
       printf("CopySourceVersionId:%s\n", $resp['CopySourceVersionId']);
       printf("LastModified:%s\n", $resp['LastModified']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

5.4 DELETE Object

API Description

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

Method Definition

```
1. ObsClient->deleteObject(array $parameter)
1. ObsClient->deleteObject(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
VersionId	string	Optional	Version ID of the object to be deleted

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
DeleteMarker	boolean	Whether the deleted object is a delete marker
VersionId	string	Version ID of the object to be deleted

Sample Code

```
try{
       $resp = $obsClient -> deleteObject([
              'Bucket' => 'bucketname',
              'Key' => 'objectkey'
       ]);
       printf("RequestId:%s\n", $resp['RequestId']);
}catch (Obs\Common\ObsException $obsException){
       printf("ExceptionCode:%s\n", $obsException->getExceptionCode());
       printf("ExceptionMessage:%s\n", $obsException->getExceptionMessage());
}
```

5.5 DELETE Objects

API Description

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

Method Definition

```
1. ObsClient->deleteObjects(array $parameter)
2. ObsClient->deleteObjectsAsync(array $parameter, callable $callback)
```

Request Parameter

Field		Type	Optional or Mandatory	Description
Bucket		string	Mandatory	Bucket name
Objects		indexed array	Mandatory	List of objects to be deleted
	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

Field		Type	Description
HttpStatusCode		integer	HTTP status code
Reason		string	Reason description
RequestId		string	Request ID returned by the OBS server
Deleteds		indexed array	List of successfully deleted objects
	Key	string	Object name
	VersionId	string	Object version ID
	DeleteMarker	boolean	Whether the deleted object is a delete marker
	DeleteMarkerVersionId	string	Version ID of the delete marker
Errors		indexed array	List of objects failed to be deleted
	Key	string	Object name

Field		Type	Description
	VersionId	string	Object version ID
	Code	string	Error code of the deletion failure
	Message	string	Error message of the deletion failure

Sample Code

```
try {
	$resp = $obsClient->deleteObjects ( [
		'Bucket' => 'bucketname',
		'Quiet' => false,
		'Objects' => [
			[
				'Key' => 'objectkey1',
				'VersionId' => null
			],
			[
				'Key' => 'objectkey2',
				'VersionId' => null
			]
		]
	] );
	printf ( "RequestId:%s\n", $resp ['RequestId'] );
	printf ( "Deleteds:\n" );
	foreach ( $resp ['Deleteds'] as $index => $deleted ) {
		printf ( "Deleteds[%d]", $index + 1 );
		printf ( "Key:%s\n", $deleted ['Key'] );
		printf ( "VersionId:%s\n", $deleted ['VersionId'] );
		printf ( "DeleteMarker:%s\n", $deleted ['DeleteMarker'] );
		printf ( "DeleteMarkerVersionId:%s\n", $deleted ['DeleteMarkerVersionId'] );
	}
	printf ( "Errors:\n" );
	foreach ( $resp ['Errors'] as $index => $error ) {
		printf ( "Errors[%d]", $index + 1 );
		printf ( "Key:%s\n", $error ['Key'] );
		printf ( "VersionId:%s\n", $$error ['VersionId'] );
		printf ( "Code:%s\n", $error ['Code'] );
		printf ( "Message:%s\n", $error ['Message'] );
	}
} catch ( Obs\Common\ObsException $obsException ) {
	printf("StatusCode:%s\n", $obsException->getStatusCode());
}
```

5.6 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

```
1. ObsClient->getObjectMetadata(array $parameter)
2. ObsClient->getObjectMetadataAsync(array $parameter, callable $callback)
```

Request Parameter

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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
LastModified	string	Time when the last modification was made to the object
ContentLength	integer	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.

Field	Type	Description
StorageClass	string	Storage class of the object. When the storage class is OBS Standard, the value is null.
Restore	string	Restore status of the object in the OBS Archive storage class
AllowOrigin	string	If **Origin** in the request meets the CORS rules of the bucket, **AllowedOrigin** in the CORS rules is returned.
AllowHeader	string	If **RequestHeader** in the request meets the CORS rules of the bucket, **AllowedHeader** in the CORS rules is returned.
AllowMethod	string	**AllowedMethod** in the CORS rules of the bucket
ExposeHeader	string	**ExposeHeader** in the CORS rules of the bucket
MaxAgeSeconds	integer	**MaxAgeSeconds** in the CORS rules of the bucket
SseKms	string	Algorithm used in SSE-KMS decryption
SseKmsKey	string	Master key used in SSE-KMS decryption
SseC	string	Algorithm used in SSE-C decryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C decryption
Expiration	string	Expiration details
Metadata	associative array	Customized metadata of the object

Sample Code

```
try {
	$resp = $obsClient->getObjectMetadata( [
		'Bucket' => 'bucketname',
		'Key' => 'objectkey'
	] );
	printf ( "RequestId:%s\n", $resp ['RequestId'] );
	printf ( "ETag:%s\n", $resp ['ETag'] );
	printf ( "VersionId:%s\n", $resp ['VersionId'] );
	printf ( "ContentLength:%s\n", $resp ['ContentLength'] );
	printf ( "LastModified:%s\n", $resp ['LastModified'] );
	printf ( "Expiration:%s\n", $resp ['Expiration'] );
	printf ( "StorageClass:%s\n", $resp ['StorageClass'] );
} catch ( Obs\Common\ObsException $obsException ) {
	printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
	printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.7 PUT Object acl

API Description

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

Method Definition

```
1. ObsClient->setObjectAcl(array $parameter)
2. ObsClient->setObjectAclAsync(array $parameter, callable $callback)
```

Request Parameter

Field			Type	Optional or Mandatory	Description
Bucket			string	Mandatory	Bucket name
Key			string	Mandatory	Object name
VersionId			string	Optional	Object version ID
ACL			string	Optional	**Pre-defined access control policy**
Owner			associative array	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			indexed array	Optional	List of grantees' permission information
	Grantee		Object	Mandatory	Grantee
		Type	string	Mandatory	**Grantee type**

Field			Type	Optional or Mandatory	Description
		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**
	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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
try {
      $resp = $obsClient->setObjectAcl( [
                    'Bucket' => 'bucketname',
                    'Key' => 'objectkey',
                    'Owner' => ['ID' => 'ownerid'],
                    'Grants' => [
                          ['Grantee' => ['Type' => 'CanonicalUser', 'ID' => 'userid'], 'Permission' =>
ObsClient::PermissionRead],
                          ['Grantee' => ['Type' => 'CanonicalUser', 'ID' => 'userid'], 'Permission' =>
ObsClient::PermissionWriteAcp],
                          ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupAuthenticatedUsers],
'Permission' => ObsClient::PermissionWriteAcp],
                          ['Grantee' => ['Type' => 'Group', 'URI' => ObsClient::GroupAuthenticatedUsers],
'Permission' => ObsClient::PermissionRead],
                    ]
       ] );
       printf ( "RequestId:%s\n", $resp ['RequestId'] );
} catch ( Obs\Common\ObsException $obsException ) {
       printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
       printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.8 GET Object acl

API Description

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

Method Definition

```
1. ObsClient->getObjectAcl(array $parameter)
2. ObsClient->getObjectAclAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Field			Type	Description
VersionId			string	Object version ID
Owner			associative array	Object owner
	ID		string	ID of the domain to which the object owner belongs
Delivered			boolean	Whether the bucket ACL is applied to objects in the bucket
Grants			indexed array	List of grantees' permission information
	Grantee		associative array	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

```
try {
        $resp = $obsClient->getObjectAcl( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey'
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
        printf("Owner[ID]:%s\n", $resp['Owner']['ID']);
        printf("Grants\n");
        foreach ($resp['Grants'] as $index => $grant){
                printf("Grants[%d]", $index + 1);
                printf("Grantee[ID]:%s\n", $grant['Grantee']['ID']);
                printf("Grantee[URI]:%s\n", $grant['Grantee']['URI']);
                printf("Permission:%s\n", $grant['Permission']);
        }
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.9 Initiate Multipart Upload

API Description

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

Method Definition

```
1. ObsClient->initiateMultipartUpload(array $parameter)
2. ObsClient->initiateMultipartUploadAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
ACL	string	Optional	**Pre-defined access control policy**
StorageClass	string	Optional	**Storage class** of the object
Metadata	associative array	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.
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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Field	Type	Description
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
SseKms	string	Algorithm used in SSE-KMS encryption
SseKmsKey	string	Key used in SSE-KMS encryption
SseC	string	Algorithm used in SSE-C encryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C encryption

Sample Code

```
try {
        $resp = $obsClient->initiateMultipartUpload( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'ContentType' => 'text/plain'
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
        printf ( "Bucket:%s\n", $resp ['Bucket'] );
        printf ( "Key:%s\n", $resp ['Key'] );
        printf ( "UploadId:%s\n", $resp ['UploadId'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.10 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 lastly being uploaded whose size ranging 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

```
1. ObsClient->uploadPart(array $parameter)
2. ObsClient->uploadPartAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
PartNumber	integer	Mandatory	Part number, which ranges from 1 to 10000
UploadId	string	Mandatory	Multipart upload ID
ContentMD5	string	Optional	Base64-encoded MD5 value of the part to be uploaded. It is provided for the OBS server to verify data integrity.
Body	string or resource or GuzzleHttp\Psr7\StreamInterface	Optional	Part content to be uploaded
SourceFile	string	Optional	Path to the source file of the part
Offset	integer	Optional	Start offset (in bytes) of a part in the source file. The default value is **0**.
PartSize	integer	Optional	Size (in bytes) of a part in the source file. The default value is the file size minus **Offset**. Except for the part lastly being uploaded whose size ranging from 0 to 5 GB, sizes of the other parts range from 100 KB to 5 GB.
SseC	string	Optional	Algorithm used in SSE-C encryption. The value can be: ● AES256
SseCKey	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

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
ETag	string	ETag of the uploaded part
SseKms	string	Algorithm used in SSE-KMS encryption
SseKmsKey	string	Key used in SSE-KMS encryption
SseC	string	Algorithm used in SSE-C encryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C encryption

Sample Code

```
try {
        $resp = $obsClient->uploadPart( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'UploadId' => 'uploadid',
                'PartNumber' => 1,
                'Body' => 'Hello OBS'
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
        printf ( "ETag:%s\n", $resp ['ETag'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.11 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

```
1. ObsClient->copyPart(array $parameter)
2. ObsClient->copyPartAsync(array $parameter, callable $callback)
```

Request Parameter

Field	Type	Optional or Mandatory	Description
Bucket	string	Mandatory	Bucket name
Key	string	Mandatory	Object name
PartNumber	integer	Mandatory	Part number, which ranges from 1 to 10000
UploadId	string	Mandatory	Multipart upload ID
CopySource	string	Mandatory	Parameter used to specify the source bucket, source object, and source object version ID which can be null. It is in the format of *SourceBucketName/ SourceObjectName?versionId=SourceObjectVersionId*.
CopySourceRange	string	Optional	Copy 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

Field	Type	Optional or Mandatory	Description
CopySourceSseCKey	string	Optional	Key used to decrypt the source object in SSE-C mode, which is calculated by using AES-256

Returned Result

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

Sample Code

```
try {
        $resp = $obsClient->copyPart( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'UploadId' => 'uploadid',
                'PartNumber' => 1,
                'CopySource' => 'sourcebucketname/sourceobjectkey',
                'CopySourceRange' => 'bytes=0-10'
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
        printf ( "ETag:%s\n", $resp ['ETag'] );
        printf ( "LastModified:%s\n", $resp ['LastModified'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.12 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

```
1. ObsClient->listParts(array $parameter)
2. ObsClient->listPartsAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server
Bucket	string	Bucket name
Key	string	Object name
UploadId	string	Multipart upload ID

Field		Type	Description
PartNumberMarker		string	Part number after which the listing uploaded parts begins, which is consistent with that set in the request
NextPartNumberMark-er		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		boolean	Whether all uploaded parts are returned. If the field value is **true**, not all uploaded parts are returned. If the field value is **false**, all uploaded parts are returned.
Parts		indexed array	List of uploaded parts
	PartNumber	integer	Part number
	LastModified	string	Time when the part was last modified
	ETag	string	Part ETag
	Size	integer	Part size
Initiator		associative array	Initiator of the multipart upload
	ID	string	ID of the domain to which the initiator belongs
	DisplayName	string	Initiator name
Owner		associative array	Owner of the multipart upload, which is consistent with **Initiator**
	ID	string	ID of the domain to which the initiator belongs
	DisplayName	string	Initiator name
StorageClass		string	Storage class of the object to be uploaded

Sample Code

```
try {
      $resp = $obsClient->listParts( [
            'Bucket' => 'bucketname',
            'Key' => 'objectkey',
            'UploadId' => 'uploadid',
            'MaxParts' => 10
      ] );
      printf ( "RequestId:%s\n", $resp ['RequestId'] );
```

```
        printf ( "Initiator[ID]:%s\n", $resp ['Initiator']['ID'] );
        printf ( "Initiator[DisplayName]:%s\n", $resp ['Initiator']['DisplayName'] );
        foreach ($resp['Parts'] as $index => $part){
                printf("Parts[%d]\n", $index + 1);
                printf ( "PartNumber:%s\n", $part['PartNumber'] );
                printf ( "LastModified:%s\n", $part['LastModified'] );
                printf ( "ETag:%s\n", $part['ETag'] );
                printf ( "Size:%s\n", $part['Size'] );
        }
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.13 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

```
1. ObsClient->completeMultipartUpload(array $parameter)
2. ObsClient->completeMultipartUploadAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason 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
SseKms	string	Algorithm used in SSE-KMS encryption
SseKmsKey	string	Master key used in SSE-KMS encryption
SseC	string	Algorithm used in SSE-C encryption
SseCKeyMd5	string	MD5 value of the key used in SSE-C encryption

Sample Code

```
try {
        $resp = $obsClient->completeMultipartUpload( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'UploadId' => 'uploadid',
                'Parts' => [
                        ['PartNumber' => 1, 'ETag' => 'etag1'],
                        ['PartNumber' => 2, 'ETag' => 'etag2']
                ]
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
        printf ( "Bucket:%s\n", $resp ['Bucket'] );
        printf ( "Key:%s\n", $resp ['Key'] );
        printf ( "ETag:%s\n", $resp ['ETag'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.14 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

```
1. ObsClient->abortMultipartUpload(array $parameter)
2. ObsClient->abortMultipartUploadAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	String	Request ID returned by the OBS server

Sample Code

```
try {
        $resp = $obsClient->abortMultipartUpload( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'UploadId' => 'uploadid'
        ] );
        printf ( "RequestId:%s\n", $resp ['RequestId'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

5.15 POST Object restore

API Description

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

Method Definition

```
1. ObsClient->restoreObject(array $parameter)
2. ObsClient->restoreObjectAsync(array $parameter, callable $callback)
```

Request Parameter

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

Returned Result

Field	Type	Description
HttpStatusCode	integer	HTTP status code
Reason	string	Reason description
RequestId	string	Request ID returned by the OBS server

Sample Code

```
try {
        $resp = $obsClient->restoreObject( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
            'Days' => 1,
            'Tier' => ObsClient::RestoreTierExpedited
      ] );
      printf ( "RequestId:%s\n", $resp ['RequestId'] );
} catch ( Obs\Common\ObsException $obsException ) {
      printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
      printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

6 Other APIs

6.1 Creating a Signed URL

API Description

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

Method Definition

```
ObsClient->createSignedUrl(array $parameter)
```

Request Parameter

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

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

Returned Result

Field	Type	Description
SignedUrl	string	Signed URL
ActualSignedRequestHeaders	associative array	Actual headers in the request initiated by using the signed URL

Sample Code

```
try {
    // Generate a signed URL for creating a bucket.
    $resp = $obsClient->createSignedUrl( [
        'Method' => 'PUT',
        'Bucket' => 'bucketname',
        'Expires' => 3600,
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );

    // Generate a signed URL for uploading an object.
    $resp = $obsClient->createSignedUrl( [
            'Method' => 'PUT',
            'Bucket' => 'bucketname',
            'Key' => 'objectkey',
            'Expires' => 3600,
            'Headers' => ['content-type' => 'text/plain']
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );

    // Generate a signed URL for setting an object ACL.
    $resp = $obsClient->createSignedUrl( [
            'Method' => 'PUT',
            'Bucket' => 'bucketname',
            'Key' => 'objectkey',
            'Expires' => 3600,
            'SpecialParam' => 'acl',
            'Headers' => ['x-obs-acl' => 'public-read']
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );

    // Generate a signed URL for downloading an object.
    $resp = $obsClient->createSignedUrl( [
            'Method' => 'GET',
            'Bucket' => 'bucketname',
            'Key' => 'objectkey',
            'Expires' => 3600
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );

    // Generate a signed URL for deleting an object.
    $resp = $obsClient->createSignedUrl( [
            'Method' => 'DELETE',
            'Bucket' => 'bucketname',
            'Key' => 'objectkey',
            'Expires' => 3600
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );

    // Generate a signed URL for deleting a bucket.
    $resp = $obsClient->createSignedUrl( [
            'Method' => 'DELETE',
            'Bucket' => 'bucketname',
            'Expires' => 3600
    ] );
    printf ( "SignedUrl:%s\n", $resp ['SignedUrl'] );
    printf ( "ActualSignedRequestHeaders:%s\n", print_r($resp ['ActualSignedRequestHeaders'], true) );
} catch ( Obs\Common\ObsException $obsException ) {
    printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
    printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

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->createPostSignature(array $parameter)
```

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	integer	Optional	Validity period of the browser-based upload authentication, in seconds. The default value is **300**.
FormParams	associative array	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

```
try {
        $resp = $obsClient->createPostSignature( [
                'Bucket' => 'bucketname',
                'Key' => 'objectkey',
                'Expires' => 3600,
                'FormParams' => [
                        'acl' => 'public-read',
                        'content-type' => 'text/plain',
                ]
        ] );
        printf ( "Policy:%s\n", $resp ['Policy'] );
        printf ( "Signature:%s\n", $resp ['Signature'] );
} catch ( Obs\Common\ObsException $obsException ) {
        printf ( "ExceptionCode:%s\n", $obsException->getExceptionCode () );
        printf ( "ExceptionMessage:%s\n", $obsException->getExceptionMessage () );
}
```

A Change History

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