

# Alibaba Cloud Object Storage Service

## Errors and Troubleshooting

Issue: 20190816

## Legal disclaimer

---

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.



## Generic conventions

Table -1: Style conventions

| Style   | Description  | Example  |
|---|--|--|
|  | This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results. |  <b>Danger:</b><br>Resetting will result in the loss of user configuration data.                                    |
|  | This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.  |  <b>Warning:</b><br>Restarting will cause business interruption. About 10 minutes are required to restore business. |
|  | This indicates warning information, supplementary instructions, and other content that the user must understand.                           |  <b>Notice:</b><br>Take the necessary precautions to save exported data containing sensitive information.          |
|   | This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.                       |  <b>Note:</b><br>You can use Ctrl + A to select all files.  |
| >   | Multi-level menu cascade.  | Settings > Network > Set network type  |
| <b>Bold</b>   | It is used for buttons, menus, page names, and other UI elements.  | Click <b>OK</b> .  |
| Courier font  | It is used for commands.   | Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.   |
| <i>Italics</i>  | It is used for parameters and variables.   | <code>bae log list --instanceid <i>Instance_ID</i></code>  |
| [ ] or [a b]  | It indicates that it is an optional value, and only one item can be selected.  | <code>ipconfig [-all -t]</code>  |

| Style                                 | Description  | Example                            |
|---------------------------------------|--|------------------------------------|
| <code>{}</code> or <code>{a b}</code> | It indicates that it is a required value, and only one item can be selected. | <code>swich {stand   slave}</code> |



# Contents

---

- Legal disclaimer..... I
- Generic conventions..... I
- 1 OSS error response..... 1
- 2 Network connection timeout handling.....13
- 3 PostObject..... 16
- 4 OSS permission.....27
- 5 Referer..... 32
- 6 Upload callback..... 37
- 7 STS common errors and troubleshooting..... 52
- 8 OSS CORS..... 57
- 9 OSS 403..... 60

# 1 OSS error response

---

If an error occurs when you access OSS, OSS returns the error code and error message so that you can locate the problem and handle it properly.

## Response message body

If an error occurs when you access OSS, OSS returns an HTTP status code (3xx, 4xx, or 5xx) and a message body in application or XML format.

An example of the message body of a error response is as follows:

```
<? xml version = " 1 . 0 " ? >
< Error xmlns = " http : // doc . oss - cn - hangzhou . aliyuncs . com
">
  < Code >
    AccessDeni ed
  </ Code >
  < Message >
    Query - string authentica tion requires the
Signature , Expires and OSSAccessK eyId parameters
  </ Message >
  < RequestId >
    1D842BC542 5544BB
  </ RequestId >
  < HostId >
    oss - cn - hangzhou . aliyuncs . com
  </ HostId >
</ Error >
```

The message body of an error response includes the following elements:

- **Code:** The error code that OSS returns to the user
- **Message:** The detailed error message returned by OSS
- **RequestId:** The UUID that uniquely identifies a request. When you cannot solve the error, you can provide this RequestId to Alibaba Cloud OSS technical support to get help.
- **HostId:** Used to identify the accessed OSS cluster, which is the same as the Host ID carried in the user request.

For special error information elements, see specific request descriptions.

## OSS error codes

The following table lists the OSS error codes:

| HTTP status code | Error code        | Description                              | Cause and solution   |
|------------------|-------------------|--|--|
| 203              | CallbackFailed    | Upload callback fails.                   | <p>The setting or format of the callback parameters is incorrect. For example, upload callback fails because the callback parameters within the ArgumentValue is not in the valid JSON format.</p> <p>To learn the cause and troubleshooting, see <a href="#">Upload callback</a>.</p> |
| 400              | InvalidBucketName | The bucket name is invalid.              | <p>The bucket name does not conform to the naming conventions.</p> <p>For more information about the bucket naming conventions, see <a href="#">Bucket</a>.</p>  |
|                  | InvalidObjectName | The object name is invalid.              | <p>The object name does not conform to the naming conventions.</p> <p>For more information about the object naming conventions, see <a href="#">Object</a>.</p>  |
|                  | TooManyBuckets    | The number of buckets exceeds the limit. | <p>An Alibaba Cloud account can create a maximum of 30 buckets in a region.</p> <p>To adjust the limit, <a href="#">open a bucket</a>.</p>   |

| HTTP status code | Error code                        | Description   | Cause and solution   |
|------------------|-----------------------------------|---|--|
|                  | RequestIsNotMultipartContent      | The content-type of the Post request is invalid.  | <p>The content-type header in the Post request is not <code>multipart / form - data</code> .</p> <p>The content-type header in a Post request must be <code>multipart / form - data</code> and in the <code>multipart / form - data ; boundary = xxxxxx</code> format, in which <i>boundary</i> is the boundary string. For more information about troubleshooting, see <a href="#">#unique_6</a>.</p> |
|                  | RequestTimeout                    | Request timeout occurs.   | <p>The request timeout occurs because of network environment or configurations.</p> <p>For more information about troubleshooting, see <a href="#">Network connection timeout handling</a>.</p>  |
|                  | NotImplemented                    | The method cannot be implemented.   | <p>This error occurs because parameters are incorrectly passed when the API is encapsulated.</p> <p>For more information about troubleshooting, see the parameters described in <a href="#">API overview</a>.</p>  |
|                  | MaxPOSTPreDataLengthExceededError | The size of the body except for the uploaded file content of the Post request is too large. | <p>The file uploaded by the Post request is larger than 5 GB.</p> <p>Only the <code>file</code> field can exceed 4 KB. For more information, see <a href="#">PostObject</a>.</p>   |

| HTTP status code | Error code                    | Description  | Cause and solution  |
|------------------|-------------------------------|--|---|
|                  | MalformedPOSTRequest          | The format of the Post request body is invalid.                  | The format of the form field is invalid.<br>For more information about troubleshooting, see <a href="#">PostObject</a> .  |
|                  | MalformedXML                  | The XML format is invalid.                                       | The XML format in the Post request is invalid.<br>For more information about troubleshooting, see the following topics: <ul style="list-style-type: none"> <li>• <a href="#">DeleteObjects</a></li> <li>• <a href="#">CompleteMultipartUpload</a></li> <li>• <a href="#">PutBucketLogging</a></li> <li>• <a href="#">PutBucketWebsite</a></li> <li>• <a href="#">PutBucketLifecycle</a></li> <li>• <a href="#">PutBucketReferer</a></li> <li>• <a href="#">PutBucketCORS</a></li> </ul> |
|                  | InvalidTargetBucketForLogging | The target bucket specified in the logging operation is invalid. | The target bucket to store logs is invalid. Specify a valid target bucket.  |
|                  | InvalidPolicyDocument         | The policy document is invalid.                                  | The policy format in the Post request is incorrect.<br>For more information about troubleshooting, see <a href="#">PostObject</a> .   |
|                  | InvalidPart                   | Invalid parts exist.   | A part uploaded by <a href="#">CompleteMultipartUpload</a> is invalid because its PartNumber or ETag is incorrect.<br>For more information about troubleshooting, see <a href="#">#unique_11</a> .  |

| HTTP status code | Error code                          | Description  | Cause and solution  |
|------------------|-------------------------------------|--|---|
|                  | InvalidPartOrder                    | The part order is invalid.   | The parts submitted by CompleteMultipartUpload is invalid. Parts must be submitted in an ascending sort order of PartNumber.<br><br>For more information about troubleshooting, see <a href="#">#unique_11</a> .  |
|                  | InvalidEncryptionAlgorithmError     | The specified entropy encryption algorithm is incorrect.           | The specified value of x-oss-server-side-encryption is invalid. Only AES256 and KMS are supported.<br><br>For more information about troubleshooting, see <a href="#">PutObject</a> .   |
|                  | InvalidDigest                       | The digest is invalid.   | If the Content-MD5 header is included in the request, OSS calculates the Content-MD5 value of the request body. If the Content-MD5 values are inconsistent, this error code is returned.<br><br>For more information about troubleshooting, see <a href="#">PutObject</a> . |
|                  | InvalidTargetType                   | The type of the object that the symbol link directs to is invalid. | The object is a symbol link and the object that the link directs to is also a symbol link.  |
|                  | InvalidArgument                     | The parameter format is incorrect.                                 | The parameter format is incorrect.<br><br>For more information about the parameter format, see <a href="#">API overview</a> .   |
|                  | IncorrectNumberOfFilesInPOSTRequest | The number of files in the Post request is invalid.                | Only one file field is allowed in the form fields of a Post request.<br><br>For more information, see <a href="#">PostObject</a> .  |

| HTTP status code | Error code         | Description                                      | Cause and solution   |
|------------------|--------------------|--|--|
|                  | FilePartNotExist   | The file part does not exist.                    | The part submitted by CompleteMultipartUpload is not uploaded.<br><br>For more information about troubleshooting, see <a href="#">#unique_11</a> .   |
|                  | FieldItemTooLong   | The form field in the Post request is too large. | Only the file field can exceed 4 KB.<br><br>For more information, see <a href="#">PostObject</a> .   |
|                  | EntityTooSmall     | The entity is too small.                         | Set the Post policy to specify the valid values of form fields when using PostObject to upload files. For example, content-length-range can be used to set the maximum and minimum size of an uploaded object. The condition supports the matching method of content-length-range, that is, the error is reported when the value is extremely large or small,<br><br>For more information about troubleshooting, see <a href="#">#unique_6</a> . |
|                  | EntityTooLarge     | The entity is too large.                         |  |
| 403              | AccessDenied       | The access is denied.                            | You do not have the permission to perform the operation.<br><br>For more information, see <a href="#">OSS permission</a> .   |
|                  | InvalidAccessKeyId | The AccessKeyId is invalid.                      | The AccessKeyId is invalid or expired.<br><br>For more information, see <a href="#">OSS 403</a> .  |

| HTTP status code | Error code            | Description  | Cause and solution  |
|------------------|-----------------------|--|---|
|                  | InvalidObjectState    | The object state is invalid.   | <p>When you download an object of the Archive class, the state of the object is invalid in the following two conditions:</p> <ul style="list-style-type: none"> <li>· The RestoreObject request is not submitted or the last RestoreObject request is timeout.</li> <li>· The RestoreObject request is submitted but the RestoreObject operation is not complete.</li> </ul> <p>For more information about troubleshooting, see <a href="#">#unique_20</a>.</p> |
|                  | RequestTimeTooSkewed  | The time when OSS receives the request is more than 15 minutes later than the time when the request is sent. | <p>Check the system time of the device from where the request is sent, and then adjust the time according to your time zone.</p> <p>For more information, see <a href="#">OSS 403</a>.</p>  |
|                  | SignatureDoesNotMatch | The signature is incorrect.  | The signature of the request is incorrect.  |
| 404              | SymlinkTargetNotExist | The target object that the symbol link directs to does not exist.  | The object is a symbol link, and the target object that the symbol link directs to does not exist.  |
|                  | NoSuchBucket          | The bucket does not exist.   | The requested bucket does not exist.  |
|                  | NoSuchKey             | The object does not exist.   | The requested object does not exist.  |
|                  | NoSuchUpload          | The ID of the MultipartUpload task does not exist.   | <p>The MultipartUpload task is not initialized or the initialized MultipartUpload task is expired.</p> <p>For more information about troubleshooting, see <a href="#">#unique_21</a>.</p>   |

| HTTP status code | Error code          | Description                   | Cause and solution  |
|------------------|---------------------|-------------------------------|---|
| 405              | MethodNotAllowed    | The method is not supported.  | The operation is not supported.   |
| 409              | BucketAlreadyExists | The bucket already exists.    | The specified bucket name already exists. Specify another name for the new bucket.<br><br>For more information about bucket naming conventions, see <a href="#">Bucket</a> .                        |
|                  | BucketNotEmpty      | The bucket is not empty.      | The bucket to be deleted includes objects that are not deleted or multipart upload tasks that are not complete.<br><br>For more information about troubleshooting, see <a href="#">#unique_22</a> . |
|                  | ObjectNotAppendable | The object is not appendable. | OSS objects can be classified into three types: normal, appendable, and multipart. The AppendObject operation can be performed only on objects of the appendable type.                              |

| HTTP status code | Error code               | Description  | Cause and solution  |
|------------------|--------------------------|--|---|
|                  | PositionNotEqualToLength | The position where the object is appended does not equal to the object size. | <ul style="list-style-type: none"> <li>The value of position is inconsistent with the size of the current object.</li> </ul> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;">  <b>Note:</b><br/>           You can obtain the position for the next operation from the response header <code>x-oss-next-append-position</code>, and then send the next request. However, the same error may occur even if the value of position is set to <code>x-oss-next-append-position</code> because of the concurrency of the requests.         </div> <ul style="list-style-type: none"> <li>When the value of position is 0, if appendable objects with the same name specified in the request does not exist or the size of an appendable object with the same name is 0, the request is successful. Otherwise, the value of position and the size of the object does not match and this error code is returned.</li> </ul> |
| 411              | MissingContentLength     | The content length is not included in the request.                           | The request header is not encoded by using <b>chunked encoding</b> , and does not include the <code>Content-Length</code> parameter.  |

| HTTP status code | Error code                       | Description                                | Cause and solution  |
|------------------|----------------------------------|--|---|
| 412              | PreconditionFailed               | The pre-processing fails.                  | <ul style="list-style-type: none"> <li>The value of If-Unmodified-Since is specified, but the specified time is before the modification time of the object.</li> <li>The value of If-Match is specified, but the ETag of the original object is different from the ETag value in the request.</li> </ul> <p>For more information about troubleshooting, see <a href="#">#unique_23</a>.</p>   |
| 503              | DownloadTrafficRateLimitExceeded | The downloading traffic exceeds the limit. | The default limit of the downloading traffic through the Internet and intranet is 5 Gbit/s. To adjust the limit, <a href="#">open a ticket</a> .  |
|                  | UploadTrafficRateLimitExceeded   | The uploading traffic exceeds the limit.   | The default limit of the uploading traffic through the Internet and intranet is 5 Gbit/s. To adjust the limit, <a href="#">open a ticket</a> .  |
|                  | MetaOperationQPSLimitExceeded    | The QPS exceeds the default limit.         | <p>OSS limits the QPS for the following APIs:</p> <ul style="list-style-type: none"> <li>Service-related operations, such as GetService (ListBuckets)</li> <li>Bucket-related operations, such as PutBucket and GetBucketLifecycle</li> <li>CORS-related operations, such as PutBucketCORS and GetBucketCORS</li> <li>LiveChannel-related operations, such as PutLiveChannel and DeleteLiveChannel.</li> </ul> <p>If the QPS exceeds the limit, this error code is returned. We recommend that you perform the operation again after a few seconds.</p> |

**Note:**

For more information about OSS error code, see [OSS API error center](#).

### Common errors and troubleshooting

For more information about OSS common errors and troubleshooting, see:

- [Upload callback](#)
- [OSS 403](#)
- [PostObject](#)
- [OSS permission](#)
- [OSS CORS](#)
- [Referer](#)
- [STS](#)

For more information about common errors and troubleshooting for SDK or tools, see:

- [Java SDK:FAQ](#)
- [Node.js SDK:FAQ](#)
- [ossfs](#)
- [ossftp](#)

### Unsupported operations

If you access OSS resources by performing an operation that is not supported by OSS, the 405 Method Not Allowed error is returned.

Example of an incorrect request:

```
ABC / 1 . txt HTTP / 1 . 1
Host : bucketname . oss - cn - shanghai . aliyuncs . com
Date : Thu , 11 Aug 2016 03 : 53 : 40 GMT
Authorizat ion : signatureV alue
```

Response example:

```
HTTP / 1 . 1 405 Method Not Allowed
Server : AliyunOSS
Date : Thu , 11 Aug 2016 03 : 53 : 44 GMT
Content - Type : applicatio n / xml
Content - Length : 338
Connection : keep - alive
x - oss - request - id : 57ABF6C8BC 4D25D86CBA 5ADE
Allow : GET DELETE HEAD PUT POST OPTIONS
<? xml version =" 1 . 0 " encoding =" UTF - 8 " ? >
< Error >
```

```

< Code > MethodNotAllowed </ Code >
< Message > The specified method is not allowed against
this resource . </ Message >
< RequestId > 57ABF6C8BC 4D25D86CBA 5ADE </ RequestId >
< HostId > bucketname . oss - cn - shanghai . aliyuncs . com </
HostId >
< Method > abc </ Method >
< ResourceType > Bucket </ ResourceType >
</ Error >

```

**Note:**

If the resource to be accessed is `/bucket/`, the `ResourceType` is `bucket`. If the resource to be accessed is `/bucket/object`, the `ResourceType` is `object`.

**Unsupported parameters in supported operations**

If unsupported parameters (such as `If-Modified-Since` in `PutObject` operations) is specified in supported OSS operations, the 400 Bad Request error is returned.

**Example of an incorrect request:**

```

PUT / abc . zip HTTP / 1 . 1
Host : bucketname . oss - cn - shanghai . aliyuncs . com
Accept : */*
Date : Thu , 11 Aug 2016 01 : 44 : 50 GMT
If - Modified - Since : Thu , 11 Aug 2016 01 : 43 : 51 GMT
Content - Length : 363

```

**Response example:**

```

HTTP / 1 . 1 400 Bad Request
Server : AliyunOSS
Date : Thu , 11 Aug 2016 01 : 44 : 54 GMT
Content - Type : application / xml
Content - Length : 322
Connection : keep - alive
x - oss - request - id : 57ABD896CC B80C366955 187E
x - oss - server - time : 0
<? xml version =" 1 . 0 " encoding =" UTF - 8 " ? >
< Error >
  < Code > NotImplemented </ Code >
  < Message > A header you provided implies functionality
that is not implemented . </ Message >
  < RequestId > 57ABD896CC B80C366955 187E </ RequestId >
  < HostId > bucketname . oss - cn - shanghai . aliyuncs . com </
HostId >
  < Header > If - Modified - Since </ Header >
</ Error >

```

## 2 Network connection timeout handling

---

### Introduction

Network connection timeout is a typical problem that OSS SDK users may encounter when they upload files with the SDK. In such a case, a `ConnectionTimeout` error is reported, negatively affecting user experience.

### Troubleshooting

Possible causes are described as follows to analyze and resolve the network connection timeout problem of OSS SDK because this problem cannot be reproduced.

#### 1. Network environment

Analyze the following network link:

Mobile phone/PC --- Carrier network --- OSS server

Your network may be at an edge node of the carrier network. Therefore, the requests sent to the carrier network are more likely to fail. You can use CDN edge nodes for acceleration, reducing the dependency of mobile phones/PCs on the carrier network. The network link is as follows:

Mobile phone/PC -- Nearest CDN edge node -- Carrier network -- OSS Server

If the problem still exists and the `ConnectionTimeout` error still occurs, read the following analysis.

#### 2. Network configuration

The following code is the detailed timeout error message:

```
" Connection TimeoutErr or & errmsg = Failed to upload
  some parts with error : Connection TimeoutErr or :
Connect timeout for 60000ms , PUT https ://***.oss - cn
- hangzhou . aliyuncs . com /***/***/***. mp4 ? partNumber = 2 &
```

```
uploadId =*** - 2 ( connected : false , keepalive socket :
false ) headers : {} part_num : 2
```

The following conclusions can be drawn from the error message:

- The connection because the client does not receive an undefined response from the server in 60 seconds.
- According to CDN logs, the timeout problem occurs because the network is disconnected before a part is completely uploaded.
- In poor network conditions, the client/PC cannot receive responses from the OSS server in a long time if the file to be uploaded is too large.

Based on the preceding conclusions, we recommend the following solutions:

- Upload files with the multipart upload method and limit the maximum part size to 1 MB.
- Add a resumable mechanism to re-upload a part that fails to be uploaded.
- Increase the timeout period.

```
// Code example of multipart upload in JS SDK

let retryCount = 0 ;
let retryCount Max = 3 ;
...
const uploadFile = function uploadFile ( client ) {
  if (! uploadFile Client || Object . keys ( uploadFile
Client ). length === 0 ) {
    uploadFile Client = client ;
  }
  ...

  console . log ( `${ file . name } => ${ key }` );
  const options = {
    progress ,
    Partsize : 1000 * 1024 , // Set the part size .
    Timeout : 120000 , // Set the timeout period .
  };
  if ( currentCheckpoint ) {
    options . checkpoint = currentCheckpoint ;
  }
  return uploadFile Client . multipartUpload ( key , file ,
options ). then ( ( res ) => {
    console . log ( ' upload success : % j ', res );
    currentCheckpoint = null ;
    uploadFile Client = null ;
  } ). catch ( ( err ) => {
    if ( uploadFile Client && uploadFile Client . isCancel ( ) )
    {
      console . log ( ' stop - upload !' ) ;
    }
    else {
      console . error ( err );
      // retry
      if ( retryCount < retryCount Max ){
        retryCount ++;
        console . error ( " retryCount : " + retryCount );
      }
    }
  }
}
```

```
        uploadFile ('');  
    }  
}  
});  
};
```

## Summary

If you access OSS data with a standard OSS domain name (for example, oss-cn-hangzhou.aliyuncs.com), your access is implemented through the carrier network. In this case, a `ConnectionTimeout` error may occur in uploads due to complex network environments (such as unstable network or poor network conditions). You can try the following solutions:

- Upload files with the multipart upload method and limit the part size in a range from 100 KB to 1 MB.



**Note:**

The OSS server does not receive parts smaller than 100 KB.

- Add a resumable mechanism to re-upload a part that fails to be uploaded.



**Note:**

The mechanism is enabled in Android/iOS SDK by default and therefore no configuration is required.

- Increase the timeout period.
- Use the CDN acceleration service to accelerate data transmission in OSS.

## 3 PostObject

---

### Introduction

PostObject uploads files to OSS using forms. In Post Object, message entities are encoded in multi-form format multipart/form-data. For more information, see [RFC 2388](#). In Put Object, parameters are passed by HTTP headers, while Post Object parameters are passed as form fields of the message body.

A PostObject message consists of the header and the body. The header and the body are separated by `\ r \ n --{ boundary }`. The body consists of a series of form fields in the following format: `Content - Dispositio n : form - data ;`  
`name ="{ key }" \ r \ n \ r \ n { value } \ r \ n --{ boundary }`.

Common headers include Host, User-Agent, Content-Length, Content-Type and Content-MD5 while form fields include key, OSSAccessKeyId, Signature, Content-Disposition, object meta (x-oss-meta-\*), x-oss-security-token, other HTTP headers (Cache-Control/Content-Type/Cache-Control/Content-Type/Content-Disposition/Content-Encoding/Expires/Content-Encoding/Expires) and file. The `file` must be the last field in those form fields.

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

### PostObject common errors

The following table shows PostObject common errors:

| No. | Error  | Cause                      | Solution  |
|-----|--|----------------------------|---|
| 1   | <b>ErrorCode:</b><br>MalformedPOSTRequest<br><b>ErrorMessage:</b> The body of your POST request is not well-formed multipart/form-data | Invalid form field format. | See PostObject form field format following the table for the correct format of form fields. |

| No. | Error  | Cause   | Solution   |
|-----|--|---|--|
| 2   | <p><b>ErrorCode:</b><br/>InvalidAccessKeyId<br/><b>ErrorMessage:</b> The OSS Access Key Id You provided does not exist in our records.</p>   | <p>AccessKeyID was disabled or did not exist, the temporary user AccessKeyID was expired or the temporary user did not provide STS Token.</p> | <p>See <a href="#">Invalid AccessKeyId Troubleshooting</a> for the troubleshooting method.</p>             |
| 3   | <p><b>ErrorCode:</b><br/>AccessDenied<br/><b>ErrorMessage:</b> Invalid according to Policy: Policy expired.</p>  | <p>The expiration in the form field Id policy was expired.</p>  | <p>Adjust expiration in policy while ensuring that the format of expiration complies with ISO8601 GMT.</p> |
| 4   | <p><b>ErrorCode:</b><br/>AccessDenied<br/><b>ErrorMessage:</b> SignatureDoesNotMatch The request signature we calculated does not match the signature you provided. Check your key and signing method.</p> | <p>Incorrect signature.</p>   | <p>See <a href="#">PostObject signature</a> for the signature method.</p>                                  |
| 5   | <p><b>ErrorCode:</b> InvalidPolicyDocument<br/><b>ErrorMessage:</b> Invalid Policy: Invalid Simple-Condition: Simple-Conditions must have exactly one property specified.</p>                              | <p>The policy contains at least one condition in the request.</p>   | <p>See <a href="#">PostObject policy format</a>.</p>   |

| No. | Error   | Cause  | Solution   |
|-----|---|--|--|
| 6   | <b>ErrorCode</b><br>: InvalidPolicyDocument<br><b>ErrorMessage</b><br>: Invalid Policy<br>: Invalid JSON:<br>unknown character  | Check the format of <code>policy</code> to verify if   | " was missing and the escape character was \.                      |
| 7   | <b>ErrorCode</b><br>: InvalidPolicyDocument<br><b>ErrorMessage</b> :<br>Invalid Policy:<br>Invalid JSON: , or ] expected  | Incorrect <code>policy</code> format in the request.   | Check if , or ] was missing in policy.                             |
| 8   | <b>ErrorCode</b> :<br>AccessDenied<br><b>ErrorMessage</b> :<br>Invalid according to Policy: Policy Condition failed: [ "starts-with" , "\$key" , "user/eric / "]"       | The <code>key</code> specified by the request and that specified by <code>policy</code> do not match.          | Check the value of the form field <code>key</code> in the request. |
| 9   | <b>ErrorCode</b> :<br>AccessDenied<br><b>ErrorMessage</b> :<br>Invalid according to Policy: Policy Condition failed: [ "eq" , "\$bucket" , "mingdi-bjx" ]               | The <code>bucket</code> specified by the request and that specified by <code>policy</code> do not match.       | Check the value of <code>bucket</code> in endpoint.                |
| 10  | <b>ErrorCode</b> :<br>AccessDenied<br><b>ErrorMessage</b> :<br>Invalid according to Policy: Policy Condition failed: [ "starts-with" , "\$x-oss-meta-prop" , "prop- "]" | File metadata <code>x-oss-meta-prop</code> specified by the request and that specified by policy do not match. | Check the value of <code>x-oss-meta-prop</code> in the request.    |

| No. | Error   | Cause   | Solution  |
|-----|---|---|---|
| 11  | <b>ErrorCode:</b><br>AccessDenied<br><b>ErrorMessage:</b><br>Invalid according to Policy: Policy Condition failed: [ “eq” , “\${field}” , “\${value}” ]                     | The { field } specified in form fields and that specified by policy do not match, or that field was not specified in the request. | Check the value of { field } in the request.                      |
| 12  | <b>ErrorCode:</b><br>AccessDenied<br><b>ErrorMessage:</b> You have no right to access this object because of bucket acl.  | Current user did not have the required permission.  | See <a href="#">OSS Permission Problems and Troubleshooting</a> . |
| 13  | <b>ErrorCode:</b><br>InvalidArgument<br><b>ErrorMessage:</b><br>The bucket POST must contain the specified ‘key’ . If it is specified, please check the order of the fields | The form field does not specify key , or it is placed after the form field file .   | Add form field key or adjust orders.                              |

- PostObject form field format

For the format of PostObject requests, note the following items:

- The header must include `Content - Type : multipart / form - data ; boundary = { boundary }`.
- The header and the body are separated by `\ r \ n -- { boundary }`.
- The form field format is `Content - Dispositio n : form - data ; name = " { key } " \ r \ n \ r \ n { value } \ r \ n -- { boundary }`.
- Form field names are case-sensitive, such as policy, key, file, OSSAccessKeyId, OSSAccessKeyId, and Content-Disposition.



Notice:

The form field `file` must be the last form field.

- When the value of `bucket` is `public - read - write`, you do not have to specify the form fields `OSSAccessKeyId`, `policy`, and `Signature`. If any of `OSSAccessKeyId`, `policy`, and `Signature` is specified, the other two form fields must be specified no matter whether `bucket` is `public - read - write` or not.

The following describes an example PostObject request:

```
POST / HTTP / 1 . 1
User - Agent : Mozilla / 5 . 0 ( Windows ; U ; Windows NT 6
. 1 ; zh - CN ; rv : 1 . 9 . 2 . 6 )
Content - Type : multipart / form - data ; boundary = 9431149156
168
Host : mingdi - hz . oss - cn - hangzhou . aliyuncs . com
Accept : text / html , image / gif , image / jpeg , * ; q = . 2
, */ * ; q = . 2
Connection : keep - alive
Content - Length : 5052
-- 9431149156 168
Content - Dispositio n : form - data ; name = " key "
test - key
-- 9431149156 168
Content - Dispositio n : form - data ; name = " Content -
Dispositio n "
attachment ; filename = D : \ img \ 1 . png
-- 9431149156 168
Content - Dispositio n : form - data ; name = " OSSAccessK eyId
"
2NeL ***** j2Eb
```



**Note:**

- In the preceding sample request, `\ r \ n` shows a new line, namely a line feed. Also, this applies to the following sample requests.
- The preceding sample request is incomplete. For the complete request, see [Post Object](#).

If you have any questions, see the sample code:

- [C#](#)
- [Java](#)

- PostObject policy format

In a PostObject request, the form field `policy` is used to verify the validity of the request and it declares the conditions that must be met by the PostObject request. Specifically, those conditions are:

- UTF-8 JSON text must be encoded with base64 before being passed into the form field `policy`.
- The `policy` must include `expiration` and `conditions` where `conditions` must contain at least one item.

The following shows an example `policy` before base64 encoding.

```
{
  " expiration ": " 2018 - 01 - 01T12 : 00 : 00 . 000Z ",
  " conditions ": [
    [ " content - length - range ", 0 , 104857600 ]
  ]
}
```

`expiration` item specifies an expiration time of the request in the ISO8601 GMT time format. For example, `2018 - 01 - 01T12 : 00 : 00 . 000Z` specifies that the request must occur before 12:00 a.m. on January 1st, 2018.

PostPolicy supports the following “conditions” :

| Name                 | Description  | Example   |
|----------------------|--|---|
| bucket               | The bucket name of the uploaded file. Exact match is supported.            | { “bucket” : “johnsmith” } or [ “eq” , “\$bucket” , “johnsmith” ] |
| key                  | The name of the uploaded file. Exact match and prefix match are supported. | [ “starts-with” , “\$key” , “user/etc/ “]                         |
| content-length-range | The maximum and minimum allowed sizes of the uploaded file.                | [ “content-length-range” , 0, 104857600]                          |
| x-oss-meta-*         | The specified object meta . Exact match and prefix match are supported.    | [ “starts-with” , “\$x-oss-meta-prop” , “prop- “]                 |

| Name   | Description  | Example   |
|--|--|---|
| success_action_redirect  | The redirection URL upon successful upload. Exact match and prefix match are supported.  | [ “starts-with” , “\$success_action_redirect” , “ http :// www . aliyun . com “ ] |
| success_action_status  | The returned status code upon successful upload if success_action_redirect is not specified. Exact match and prefix match are supported. | [ “eq” , “\$success_action_status” , “204” ]                                      |
| Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires, and so on | The HTTP headers passed as form fields. Exact match and prefix match are supported.  | [ “eq” , “\$Content-Encoding” , “ZLIB” ]  |

PostPolicy supports the following escape characters and uses \ for escape.

| Escape Character | Description           |
|------------------|-----------------------|
| /                | Slash                 |
| \                | Backslash             |
| “                | Double quotation mark |
| \$               | Dollar sign           |
| \b               | Blank                 |
| \f               | Form feed             |
| \n               | Line feed             |
| \r               | Enter                 |
| \t               | Horizontal tab        |
| \uxxxx           | Unicode character     |

For more information about PostPolicy, see [Post Policy](#).

- PostObject signature

For a Post request to be verified, it must include AccessKeyID, policy, and Signature form fields. The signature calculation process is as follows:

1. Create a policy encoded with UTF - 8 .
2. Encode the policy with base64 . The resulting value is the value to be populated into the policy form field, and this value is used as the string to be signed.
3. Sign the string with AccessKeyS ecret . Specifically, hash the string with hmac-sha1 and then encode it with base64. The signature method is the same as that for [Header Signature](#).

Namely:

```
Signature = base64 ( hmac - sha1 ( AccessKeyS ecret , base64 (
policy )))
```

Specify the calculated signature in the form field Signature as follows:

```
Content - Dispositio n : form - data ; name =" Signature "
{ signature }
-- 9431149156 168
```

If you have any questions, see the sample code:

- [C#](#)
- [Java](#)

## FAQ

- How to specify a key?

The key is the object name, which is specified in the form field key . The following shows an example:

```
Content - Dispositio n : form - data ; name =" key "
{ key }
-- 9431149156 168
```

- How to specify object content?

Specify object content in the form field file . The following shows an example:

```
Content - Dispositio n : form - data ; name =" file "; filename
=" images . png "
Content - Type : image / png
{ File - content }
```

```
-- 9431149156 168
```

**Note:**

- The form field `file` must be the last field in a form, namely it must be placed after any other form fields.
- `filename` is the name of the uploaded local file but not the object name.

- How to specify `content - type` of the object?

Specify `content - type` of the object in the form field `file` but not in `content - type` of the header. The following shows an example:

```
Content - Dispositio n : form - data ; name =" file "; filename
=" images . png "
Content - Type : image / png
{ file - content }
-- 9431149156 168
```

- How to specify `content - md5` verification for object content?

Specify `content - md5` in the Post Object request header. Note that the MD5 value is for the entire body namely for all form fields. The following shows an example request header:

```
POST / HTTP / 1 . 1
User - Agent : Mozilla / 5 . 0 ( Windows ; U ; Windows NT 6
. 1 ; zh - CN ; rv : 1 . 9 . 2 . 6 )
Content - Type : multipart / form - data ; boundary =
9431149156 168
Content - MD5 : tdqHe4hT / TuKb7Y4by + nJg ==
Host : mingdi - hz . oss - cn - hangzhou . aliyuncs . com
Accept : text / html , image / gif , image / jpeg , * ; q = . 2
, */ * ; q = . 2
Connection : keep - alive
Content - Length : 5246
-- 9431149156 168
```

- How to specify a signature?

See `PostObject signature` for the signature calculation method. The signature is carried by the form field `Signature` .

- How to implement Post Object with STS Token of a temporary user?

The usage of `AccessKeyID` and `AccessKeySecret` of a temporary user key is the same as that of a master user key and sub-user key. `Token` is carried by the form field

`x - oss - security - token` . The following shows an example:

```
Content - Dispositio n : form - data ; name =" Signature "
5L0 + KaeugxYygf qWLJLoy0eh 0mA =
```



For more information about file meta information, see [File Meta Information Object Meta](#).

- How to specify conditions such as expiration, Key, Bucket, size, and header?

PostObject for OSS supports various conditions and can meet demanding security requirements. Specify conditions in the form field `policy`. The following shows an example policy:

```
" expiration ": " 2018 - 01 - 01T12 : 00 : 00 . 000Z ",
" conditions ": [
  [" eq ", "$ bucket ", " md - hz "],
  [" starts - with ", "$ key ", " md / conf /"],
  [" content - length - range ", 0 , 104857600 ]
```

In the preceding policy, the conditions for user Post Object operations are as follows:

- `bucket` must be `md - hz`.
  - `key` must be started with `md / conf /`.
  - The size of the uploaded file must be less than 100 MB.
  - The request time must be earlier than `2018 - 01 - 01T12 : 00 : 00 . 000Z`.
- How to specify HTTP headers such as Cache-Control, Content-Type, Content-Disposition, Content-Encoding and Expires?

Specify HTTP headers including `Cache - Control`, `Content - Type`, `Content - Dispositio n`, `Content - Encoding`, `Expires` in form fields.

For the meanings of those HTTP headers, see [RFC2616](#). However, `Content - MD5` needs to be specified in Post Header.

#### Post Object examples

- [C# Post Demo](#)
- [Java Post Demo](#)

#### Common links

- [Post object](#)
- [Java PostObject](#)

## 4 OSS permission

### OSS errors 403

An OSS error 403 indicates that the HTTP status code returned from OSS is 403 and that the server receives your request but rejects to provide service because you have no access permission. OSS errors 403 and causes are listed in the following table:

| Error                 | Message   | Cause   | Solution   |
|-----------------------|---|---|--|
| SignatureDoesNotMatch | <b>ErrorCode:</b><br>SignatureDoesNotMatch<br><b>ErrorMessage:</b><br>The request signature we calculated does not match the signature you provided. Check your key and signing method.   | Client and service calculated signatures do not match | <a href="#">OSS 403 errors and troubleshooting</a> |
| PostObject            | <b>ErrorCode:</b><br>AccessDenied<br><b>ErrorMessage:</b><br>Invalid according to Policy: Policy expired.<br><b>ErrorCode:</b><br>AccessDenied<br><b>ErrorMessage:</b><br>Invalid according to Policy: Policy Condition failed: ... | Invalid policy in postobject                          | <a href="#">PostObject</a>                         |

| Error        | Message  | Cause   | Solution  |
|--------------|--|---|---|
| Cors         | ErrorCode:<br>AccessForbiddenErrorMessage<br>: CORSResponse:<br>This CORS request is not allowed. This is usually because the evaluation of Origin, request method / Access-Control-Request-Method or Access-Control-Request-Headers are not whitelisted by the resource' s CORS spec. | CORS is not configured or is not configured incorrectly | <a href="#">OSS set up cross-domain access</a>  |
| Refers       | ErrorCode:<br>AccessDeniedErrorMessage<br>: You are denied by bucket referer policy.   | Check the Referer configuration for the bucket          | <a href="#">OSS Anti-leech</a>                  |
| AccessDenied | See the following permissions for common errors  | You have no permission.                                 | See the following content for more information. |

Among them, the permissions issue is part of the 403 error. The error with the permission problem is `AccessDenied`. These errors are described in detail below.

#### Common permissions errors

The privilege issue is that the current user does not have permission to specify an action. The errors returned by OSS and their causes can be found in the following table:

| SN | Error  | Cause   |
|----|--|---|
| 1  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.</b> | The bucket does not match the endpoint.   |
| 2  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: You are forbidden to list buckets.</b>  | You have no permissions for listBuckets.  |
| 3  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: You do not have write acl permission on this object</b>   | You have no permissions for setObjectAcl.   |
| 4  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: You do not have read acl permission on this object.</b>   | You have no permissions for getObjectAcl.   |
| 5  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: The bucket you access does not belong to you.</b>   | The subaccount has no permissions for bucket management like getBucketAcl, CreateBucket, deleteBucket, setBucketReferer, and getBucketReferer.          |
| 6  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: You have no right to access this object because of bucket acl.</b>  | The subaccount/ temporary account has no permissions to access the object like putObject, getObject, appendObject, deleteObject, and postObject.        |
| 7  | <b>ErrorCode: AccessDenied</b><br><b>ErrorMessage: Access denied by authorizer's policy.</b>   | The temporary account has no access permissions. The authorization policy specified for assuming the role of this temporary account has no permissions. |

| SN | Error   | Cause   |
|----|---|---|
| 8  | ErrorCode: AccessDenied<br>ErrorMessage: You have no right to access this object. | The subaccount/<br>temporary account has no permissions for the current operation like initiateMultipartUpload. |

### Permission error troubleshooting

Check whether the key is for the primary user, the subaccount or the temporary account.

- Check whether the key is for a primary user.

Log on to the [console](#) to check whether the AccessKeyID exists. If it does exist, the key is for a primary user.

- Check the subaccount permission, that is, the authorization policy.

Check the subaccount AccessKeyID and find out the corresponding subaccount by navigation to Resource Access Management > User Management > Management > User Details > User AccessKey.

Log on to the console and navigate to Resource Access Management > User Management > Management > User Authorization Policy > Individual Authorization Policy/User Authorization Policy to check the permissions.

- Check the permissions for a temporary account.

The AccessKeyID for the temporary account can be recognized easily since it starts with “STS” , for example, “STS.MpsSonrqGM8bGjR6CRKNMoHXe” . Log on to the console and navigate to Resource Access Management > Role Management > Management > Role Authorization Policy > View Permissionsto check the permissions.

The access rights error process is shown in the following figure:

Procedures for checking the permissions:

1. List the required permissions and resources.
2. Check whether Action has the required operation.
3. Check whether Resource is the required operation object.
4. Check whether Effect is “Allow” instead of “Deny” .
5. Check whether Condition is set correctly.

If it is unable to detect the error through checking, the following adjustments are required:

1. The condition, if any, must be removed.
2. Remove “Deny” in Effect.
3. Change Resource to “Resource” : “\*” .
4. Change Action to “Action” : “oss:\*” .



**Note:**

- We recommend that you use the OSS authorization policy generation tool [RAM Policy Editor](#) to generate authorization policies.
- For more information about RAM, see [access control for Alibaba Cloud](#).

## 5 Referer

---

### Introduction

`Referer` is part of an HTTP header. When your browser sends a request to a web server, the request normally carries a referer that notifies the server of the source page for the current request. The correct spelling of `Referer` must be the word `Referrer`. However, developers repeat the typo considering its massive usage.

### Referer functions

- **Anti-leech.** For example, when a website accesses its own image server, the image server obtains the referer to determine if the requesting domain name falls within its own domain names. If it is true, access is permitted; if it is false, access is denied.
- **Data statistics.** For example, the referer collects the statistics on source links of user accesses.

### Blank referer

A blank referer means that the content of the referer header in an HTTP request is empty, or the HTTP request does not contain the referer header.

The referer is blank in the following two scenarios:

- The request is not triggered by a link. For example, open a page by directly entering an address in the address bar.
- The referer won't be detected on an HTTP page if you access a non-encrypted HTTP page through the link on the HTTPS page.

What is the difference between allowing and disallowing blank referers in anti-leech settings?

In the whitelist of anti-leech settings, specifying that an item in the list can contain blank referers allows you to directly access this resource URL on the address bar of a browser. In other words, failing to do so disallows you to directly access it in a browser.

## OSS anti-leech

OSS anti-leech is implemented through `Referer`. Therefore, this function is also referred to as OSS `Refer` or `refer` for short. For more information, see [OSS anti-leech](#).

- OSS anti-leech configuration

OSS anti-leech includes:

- Permit for access requests with blank referer fields
- Whitelist of referer fields

OSS refer is configured by setting the `bucket` property on the OSS console or in the [SDK](#).

- OSS anti-leech precautions

Note the following precautions for OSS refer configuration:

- Anti-leech verification is performed only when users access objects through URL signatures or anonymously. If the request header contains the `Authorization` field, anti-leech verification is skipped.
- A bucket supports multiple referer parameters which are separated by commas (,).
- Referer parameters support wildcard characters `*` and `?`. For more information, see the description for the following wildcard characters.
- Users can set `whether access requests with blank referer fields are permitted`.
- When the whitelist is empty, no check is performed for blank referer fields (otherwise all requests would be rejected).
- When the whitelist is not empty and is set with rules for disallowing blank referer fields, only the requests with referers defined in the whitelist are

permitted while any other requests (including the requests with blank referers) are rejected.

- If the whitelist is not empty and is set with rules for allowing blank referer fields, requests with blank referers or which meet the whitelist are permitted while any other requests are rejected.
- All the three bucket permissions (private, public-read, and public-read-write) check referer fields.

Description for wildcard characters:

- Asterisk (\*): represents zero or multiple characters. If you are looking for a file that starts with AEW in the name but have forgotten the rest part, you can enter AEW\* to search for all types of files starting with AEW in the name, such as AEWI.txt, AEWU.EXE, and AEWI.dll. To narrow down the search scope, you can enter AEW\*.txt to search for all .txt files whose names start with AEW, such as AEWIP.txt and AEWDF.txt.
- Question mark (?): represents a single character. By entering love?, you can search for all types of files whose names are 'love' followed by a single ending character, such as lovey and lovei. To narrow down the search scope, you can enter love?.doc to search for all .doc files whose names are 'love' followed by a single ending character, such as lovey.doc and lovei.doc.
- Typical configuration
  - Permit accesses by all requests
    - Blank referer: allow blank referers
    - Referer list: empty
  - Only permit accesses by requests with specified referers
    - Blank referer: disallow blank referers
    - Refer list: `http ://*.oss-cn-beijing.aliyuncs.com`, `http ://*.aliyun.com`
  - Permit accesses by requests with specified referers and without referers
    - Blank referer: allow blank referers
    - Refer list: `http ://*.oss-cn-beijing.aliyuncs.com`, `http ://*.aliyun.com`

## Common errors and troubleshooting

When a referer is misconfigured, the HTTP status code (`http code`) is 403 and OSS returns the following error:

```
< Code > AccessDenied </ Code >
< Message > You are denied by bucket referer policy .</
Message >
```



### Note:

- Normally, referer error reporting applies to site applications, and you can view the referer of a header in browsers. For example, in Google Chrome, press **F12** to open **Developer Tools** and view the header of an element in **Network**.
- Errors returned by OSS can be obtained by capturing packets. For example, in Wireshark, you can specify the filter as `host bucket - name . oss - cn - beijing . aliyuncs . com`.

### Possible causes:

- The referer is blank and the request header contains no or blank referer fields.
- The referer is out of the specified referer range. Note the following items:
  - Determine whether the refer configuration is prefixed with `http ://` or `https ://` during configuration.
  - `a.aliyun.com` and `b.aliyun.com` match with `http ://*. aliyun . com` or `http ://?. aliyun . com`.
  - `domain.com` matches with `http :// domain . com` instead of `http ://*. domain . com`.
- Check that the refer configuration is always prefixed with `http ://` or `https ://`; otherwise, the referer format is incorrect and the refer configuration is invalid. For example, `b.aliyun.com` is an invalid configuration.



### Note:

- Configure referers in OSS Console >> **Bucket** >> **Bucket Properties** >> **Anti-Leech**.
- Clear browser cache for debugging.
- OSS refer supports only the whitelist but not the blacklist temporarily.

For troubleshooting other errors, see [OSS 403 Errors and Troubleshooting](#).

#### Other problems

Why can videos still be captured with curl when anti-leech is active?

Check if CDN is enabled, refer settings for CDN are nonempty and the anti-leech list for CDN is consistent with that for OSS. For anti-leech settings for CDN, see [Anti-Leech in the CDN User Guide](#). When debugging OSS referers, first eliminate the impact caused by CDN. Adjust OSS referers and then CDN referers.

For more information about referers and their configuration, see [Anti-Leech](#).

## 6 Upload callback

This topic describes common errors in callback functions in upload operations and how to handle them.

### About upload callback

When a file is uploaded, the OSS can provide a [Callback](#) to your callback server. You can carry the relevant callback parameters in the upload request to implement the upload callback. The APIs that support upload callback are [PutObject](#), [PostObject](#), and [CompleteMultipartUpload](#). For more information, see [Upload callback](#) and [Callback API](#) in the Developer Guide.



#### Note:

A callback server is also called a service server.

### Application scenario

- Notification

A typical application is to upload and callback by an authorized third party who specifies the callback parameters during file upload. After the upload is complete, the OSS sends a callback request to the callback server. When receiving the callback request, the callback server records the upload information.

- Processing, review, and statistics

When receiving a callback request, the callback server processes, reviews, and makes statistics on the uploaded files.

### Data stream

The following table describes the data streams.

| Data stream | Meaning   | Description   |
|-------------|---|---|
| 1           | The client uploads a file and carries a callback parameter. For more information about the format, see <a href="#">SDK/PostObject</a> . | The upload is implemented by SDK ( <a href="#">PutObject</a> and <a href="#">CompleteMultipartUpload</a> ), and the callback by the <a href="#">PostObject</a> API. |

| Data stream | Meaning  | Description  |
|-------------|--|--|
| 2           | The OSS instance stores the file and initiates a callback. | <p>The OSS instance sends a <code>POST</code> request to the specified <code>CallbackUrl</code> in the upload request. The callback time-out period is five seconds, which is a fixed value and cannot be configured.</p> <p>For more information about the format of the <code>POST</code> request, see <a href="#">Initiate a callback request</a>.</p>                  |
| 3           | The callback server returns the processing result.         | <ul style="list-style-type: none"> <li>The message body returned by the callback server must be in JSON format.</li> <li>OSS determines that the callback fails if the returned result is not the 200 status code. The <code>40x</code> code indicates invalid parameters or callback failures. The <code>50x</code> indicates time-out or connection failures.</li> </ul> |
| 4           | The OSS returns the upload and callback result.            | <ul style="list-style-type: none"> <li>If both the upload and callback succeed, <code>200</code> is returned.</li> <li>If the upload succeeds but the callback fails, <code>203</code> is returned. The value of <code>ErrorCode</code> is <code>CallbackFailed</code>, and <code>ErrorMessage</code> indicates the error cause.</li> </ul>                                |

### SDK/PostObject

During the file upload, you can set the callback parameters to specify the URL of the callback server, data to be sent to the callback server, and data format. When the callback server processes a callback, some context information, such as the `bucket` and `object`, is specified using system variables. Other context information is specified using custom variables.

The following parameters are available for an upload callback:

| Field                    | Meaning                 | Description |
|--------------------------|-------------------------|-------------|
| <code>callbackUrl</code> | Callback server address | Required    |

| Field            | Meaning   | Description   |
|------------------|---|---|
| callbackHost     | Value of the <code>Host</code> in the callback request message header   | Optional. The default value is <code>callbackUrl</code> .   |
| callbackBody     | Callback request message body   | Required. It can hold system variables and custom variables.  |
| callbackBodyType | Value of <code>Content - Type</code> in the callback request message header, that is, the <code>callbackBody</code> data format | Optional. It can be <code>application / x - www - form - urlencoded</code> (default) or <code>application / json</code> . |

Upload callback parameters are carried by the upload request in either of the following two ways:

- The callback parameters are carried by `x - oss - callback` in the message header. This is a common and recommended way.
- The callback parameters are carried by `callback` in QueryString.

Rules for generating the `x - oss - callback` or `callback` values are as follows:

```

Callback := Base64 ( CallbackJs on )
CallbackJs on := '{' CallbackUrlItem , CallbackBodyItem [,
CallbackHostItem , CallbackBodyTypeItem ]}'
CallbackUrlItem := '" callbackUrl "' ':' '" CallbackUrl
Value "'
CallbackBodyItem := '" callbackBody "' ':' '" CallbackBo
dyValue "'
CallbackHostItem := '" callbackHost "' ':' '" CallbackHo
stValue "'
CallbackBodyTypeItem := '" callbackBodyType "' : '"'
CallbackBodyType := application / x - www - form - urlencoded
| application / json

```

`CallbackJs on` value examples are as follows:

```

" callbackUrl " : " http :// abc . com / test . php ",
" callbackHost " : " oss - cn - hangzhou . aliyuncs . com ",
" callbackBody " : "{ \" bucket \":${ mimeType }, \" object \":${
object }, \" size \":${ size }, \" mimeType \":${ mimeType }, \" my_var
\":${ x : my_var }}"
" callbackBodyType " : " applicatio n / json "

```

or

```
" callbackUrl " : " http :// abc . com / test . php ",
" callbackBody " : " bucket =${ bucket }& object =${ object }&
etag =${ etag }& size =${ size }& mimeType =${ mimeType }& my_var =${
x : my_var }"
```

### System variables and custom variables

Variables for `CallbackJs on`, such as `${ bucket }`, `${ object }`, and `${ size }`, in the `CallbackJs on` example are the OSS-defined system variables. During the callback, the OSS replaces the system variables with actual values. The following table lists the OSS-defined system variables.

| Variable                          | Meaning                             |
|-----------------------------------|-------------------------------------|
| <code>\${bucket}</code>           | Storage space name                  |
| <code>\${object}</code>           | File name                           |
| <code>\${etag}</code>             | File' s etag                        |
| <code>\${size}</code>             | File size                           |
| <code>\${mimeType}</code>         | File type, such as image/jpeg       |
| <code>\${imageInfo.height}</code> | Image height                        |
| <code>\${imageInfo.width}</code>  | Image width                         |
| <code>\${imageInfo.format}</code> | Image format, such as .jpg and .png |



#### Note:

- The system variables are case sensitive.
- The system variable is in the `${ bucket }` format.
- `imageInfo` is set for images. For the non-image format, the value of `imageInfo` is blank.

Variables for `CallbackJs on`, such as `${ x : my_var }`, in the `CallbackJs on` example are the custom variables. During the callback, the OSS replaces the custom variables with custom values. Custom variable values are defined and carried by the upload request in either of the following two ways:

- The custom variables are carried by `x - oss - callback - var` in the message header. This is a common and recommended way.
- The custom variables are carried by `callback - var` in `QueryString`.

Rules for generating the `x - oss - callback - var` or `callback - var` values are as follows:

```
CallbackVa r := Base64 ( CallbackVa rJson )
CallbackVa rJson := '{' CallbackVa rItem [, CallbackVa rItem
]* '}'
CallbackVa rItem := '""' x ':' VarName '""' : '""' VarValue '""'
```

`CallbackVa rJson` value examples are as follows:

```
" x : my_var1 " : " value1 ",
" x : my_var2 " : " value2 "
```



#### Note:

- The custom variables must start with `x:`. They are case sensitive and in the format of `${ x : my_var }`.
- The custom variable length is limited by the length of the message header and URL. We recommend that the number of the custom variables do not exceed 10 and the total length do not exceed 512 bytes.

#### SDK usage example

Some SDKs, such as JAVA and JS, encapsulate the preceding steps. Some SDKs, such as Python, PHP, and C, need to use the preceding rules to generate the upload callback parameters and custom variables. The following table lists SDK usage examples.

| SDK    | Upload callback example             | Description:   |
|--------|-------------------------------------|--|
| JAVA   | <a href="#">CallbackSample.java</a> | Note the escape characters in <code>CallbackBo dy</code> . |
| Python | <a href="#">object_callback.py</a>  | -  |

| SDK     | Upload callback example                            | Description:   |
|---------|--|--|
| PHP     | <a href="#">Callback.php</a>                       | <code>OSS_CALLBACK</code> and <code>OSS_CALLBACK_VAR</code> in \$options do not need to be encoded using Base64, which is implemented by the SDK.                |
| C #     | <a href="#">UploadCallbackSample.cs</a>            | Use <code>using</code> to read <code>to</code> <code>read</code> , <code>PutObjectResult</code> . <code>ResponseStream</code> but make sure that it is disabled. |
| JS      | <a href="#">object.test.js</a>                     | -  |
| C       | <a href="#">oss_callback_sample.c</a>              | -  |
| Ruby    | <a href="#">callback.rb</a>                        | -  |
| iOS     | <a href="#">Callback notification after upload</a> | Make sure that <code>&lt; var1 &gt;</code> the format of <code>var1</code> is <code>x:var1</code> .  |
| Andriod | <a href="#">Callback notification after upload</a> | Note the escape characters in <code>CallbackBody</code> .  |



Note:

The Go SDK does not support upload callback currently.

### PostObject usage example

PostObject supports the upload callback, whose callback parameters are carried by the form field `callback` and custom variables are carried by an independent form field. For more information, see [PostObject](#).

The following table lists PostObject usage examples.

| SDK    | Upload callback example               |
|--------|---------------------------------------|
| Java   | <a href="#">PostObjectSample.java</a> |
| Python | <a href="#">object_post.py</a>        |
| C#     | <a href="#">PostPolicySample.cs</a>   |

## Callback server

The callback server is an HTTP server that processes callback requests and POST messages sent from the OSS. The callback server URL is the value of the upload callback parameter `callbackUrl`. You can implement your own processing logic on the callback server for recording, review, processing, and statistics of the uploaded data.

### Callback signature

The callback server needs to verify the signature of a POST request to make sure that the POST request is from the OSS upload callback. The callback server also can directly process the message without verifying the signature. To enhance the security of the callback server, we recommend that the callback server verify the message signature. For more information about the callback signature rules, see [Callback signature](#).



#### Note:

The OSS callback server example describes how to implement signature verification. We recommend that you directly use the code.

### Message processing

The main logic of the callback server is to process the OSS callback request. Note the following items:

- The callback server must process the POST request of the OSS.
- The OSS callback time-out time is five seconds. Therefore, the callback server must complete processing within five seconds and return the result.
- The message body sent from the callback server to the OSS must be in JSON format.
- The callback server uses its own logic, and the OSS provides examples instead of the specific service logic.

### Implementation example

The following table describes the implementation examples of the callback server.

| Language | Example                                    | Running method  |
|----------|--|---|
| JAVA     | <a href="#">AppCallbackServer.zip</a>      | Decompress the package and run <code>java -jar oss - callback - server - demo . jar 9000 .</code> |
| PHP      | <a href="#">callback-php-demo.zip</a>      | Deploy and run the program to in Apache environment.  |
| Python   | <a href="#">callback_app_server.py.zip</a> | Decompress the package and run <code>python callback_app_server . py .</code>                     |
| Ruby     | <a href="#">oss-callback-server</a>        | Run <code>ruby aliyun_oss _callback_ server . rb .</code>   |

### Debugging procedure

The upload callback debugging includes debugging of the client that uploads a file and the callback server that processes the callback. We recommend that you debug the client first and then the callback server. After independently debugging the two parts, perform the complete upload callback.

- Client debugging

You can use the callback server `http://oss-demo.aliyuncs.com:23450` provided by the OSS, that is, the callback parameter `callbackUrl` to debug the client. The callback server only verifies the callback request signature, and does not process the callback request. For callback requests whose signatures are successfully verified, the callback server returns `{"Status": "OK"}`. For callback requests whose signatures fail to be verified, the callback server returns `400 Bad Request`. For non-POST requests, the callback server returns `501 Unsupported method`. For more information about the code of the callback server example, see [callback\\_app\\_server.py.zip](#).

- Callback server debugging

The callback server is an HTTP server that can process the POST request. You can modify the callback server based on the example provided by the OSS or

implement it by yourself. The following table describes the examples of the callback server provided by the OSS.

| Language | Example                                    | Running method   |
|----------|--|--|
| JAVA     | <a href="#">AppCallbackServer.zip</a>      | Decompress the package and run <code>java -jar oss - callback - server - demo . jar 9000 .</code>          |
| PHP      | <a href="#">callback-php-demo.zip</a>      | Deploy and run the program to in Apache environment  |
| Python   | <a href="#">callback_app_server.py.zip</a> | Decompress the package and run <code>python callback_a pp_server . py .</code>                             |
| C#       | <a href="#">callback-server-dotnet.zip</a> | Compile the program and run <code>aliyun - oss - net - callback - server . exe 127 . 0 . 0 . 1 80 .</code> |
| Go       | <a href="#">callback-server-go.zip</a>     | Compile the program and run <code>aliyun_oss _callback_ server .</code>                                    |
| Ruby     | <a href="#">oss-callback-server</a>        | Run <code>ruby aliyun_oss _callback_ server . rb .</code>  |

The callback server can be debugged by running the `cURL` command. The following commands may be used:

```
# Run the following command to send a `POST` request whose message body is `object = test_obj` to the callback server :
curl -d "object = test_obj" http://oss - demo . aliyuncs . com : 23450 -v
# Run the following command to send a `POST` request whose message body is `post . txt` to the callback server :
curl -d @post . txt http://oss - demo . aliyuncs . com : 23450 -v
# Run the following command to send a `POST` request whose message body is `post . txt` and which
```

```

carries the specified message header `Content - Type`
to the callback server :
curl -d @post.txt -H "Content - Type : applicatio n /
json" http://oss-demo.aliyuncs.com:23450 -v

```



#### Note:

- When debugging the callback server, ignore signature verification because it is difficult for `cURL` to simulate the signature function.
- The OSS example already provides the signature verification function. We recommend that you directly use it.
- We recommend that the callback server provide the logging function to record all messages, facilitating debugging and tracking.
- After correctly processing a callback request, the callback server must return `200` instead of `20x`.
- The message body sent from the callback server to the OSS must be in JSON format, and `Content - Type` is set to `applicatio n / json`.

### Common errors and causes

- **InvalidArgument**

```

< Error >
  < Code > InvalidArgument </ Code >
  < Message > The callback configuration is not json
  format.</ Message >
  < RequestId > 587C79A3DD 373E2676F7 3ECE </ RequestId >
  < HostId > bucket.oss-cn-hangzhou.aliyuncs.com </ HostId
  >
  < ArgumentName > callback </ ArgumentName >
  < ArgumentValue >{"callbackUrl":"8.8.8.8:9090","
  callbackBody":{"bucket":"${bucket}","object":"${object
  }"},"callbackBodyType":"applicatio n / json"}</ ArgumentVa
  lue >
</ Error >

```



#### Note:

The callback parameter settings are incorrect, or the parameter format is incorrect. The common error is that the callback parameters in `ArgumentValue` are not in valid JSON format. In JSON, `\` and `"` are escape characters. For example, `"callbackBody":{"bucket":"${bucket}","object":"${object}"}` must be `"callbackBody":{"\bucket \":"${bucket} ,`

`\\" object \":${ object }}`". For more information about the SDKs, see the upload callback examples in the [SDK usage example](#) part.

| Character after escape | Character before escape |
|------------------------|-------------------------|
| \\                     | \\\\                    |
| “                      | \\”                     |
| \\b                    | \\b                     |
| \\f                    | \\f                     |
| \\n                    | \\n                     |
| \\r                    | \\r                     |
| \\t                    | \\t                     |

- **CallbackFailed**

Examples of **CallbackFailed** error are described as follows:

- **Example 1**

```
< Error >
  < Code > CallbackFailed </ Code >
  < Message > Response body is not valid json format
.</ Message >
  < RequestId > 587C81A125 F797621829 923D </ RequestId >
  < HostId > bucket . oss - cn - hangzhou . aliyuncs . com </
HostId >
</ Error >
```



**Note:**

The message body sent from the callback server to the OSS is not in JSON format. You can confirm the content by running `curl -d "< Content >" < CallbackServerURL > -v` or capture packets. We recommend that you use Wireshark to capture packets in Windows, and use tcpdump to capture packets in Linux. Invalid returned messages include: `OK` and `\ 357 \ 273 \ 277 {" Status ":" OK "}` (the BOM header containing the `ef bb bf` bytes).

- **Example 2**

```
< Error >
  < Code > CallbackFailed </ Code >
  < Message > Error status :-1 . OSS can not connect
to your callbackUrl , please check it .</ Message >
  < RequestId > 587C873535 5BE8694A8E 9100 </ RequestId >
  < HostId > bucket . oss - cn - hangzhou . aliyuncs . com </
HostId >
```

```
</ Error >
```

**Note:**

The processing time of the callback server exceeds five seconds. Therefore, the OSS determines that a time-out occurs. We recommend that you modify the processing logic of the callback server to asynchronous processing to make sure that it can complete processing within five seconds and returns the result to the OSS.

- **Example 3**

```
< Error >
  < Code > CallbackFa iled </ Code >
  < Message > error status :- 1 8 . 8 . 8 . 8 : 9090 reply
    timeout , cost : 5000 MS , timeout : 5000 MS ( Ernest
- 4 , errno170 ) </ message >
  < RequestId > 587C8D382A E0B92FA3EE F62C </ RequestId >
  < HostId > bucket . oss - cn - hangzhou . aliyuncs . com </
HostId >
</ Error >
```

**Note:**

The processing time of the callback server exceeds five seconds. Therefore, the OSS determines that a time-out occurs.

- **Example 4**

```
< Error >
  < Code > CallbackFa iled </ Code >
  < Message > Error status : 400 .</ Message >
  < RequestId > 587C89A02A E0B92FA3C7 981D </ RequestId >
  < HostId > bucket . oss - cn - hangzhou . aliyuncs . com </
HostId >
</ Error >
```

**Note:**

The status code of the message sent from the callback server to the OSS is 400  
 . Check the processing logic of the callback server.

- **Example 5**

```
< Error >
  < Code > CallbackFa iled </ Code >
  < Message > Error status : 502 .</ Message >
  < RequestId > 587C8D382A E0B92FA3EE F62C </ RequestId >
  < HostId > bucket . oss - cn - hangzhou . aliyuncs . com </
HostId >
```

</ Error >



**Note:**

The callback server is not started, `CallbackUrl` is missing in the callback parameters, or the network between the OSS instance and the callback server is disconnected. We recommend that you deploy the callback server on the ECS, which belongs to the same intranet as the OSS, to save the traffic cost and guarantee the network quality.

- The body of the response is not in JSON format.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>CallbackFailed</Code>
  <Message>Response body is not valid json format.</Message>
  <RequestId>5763E6EC338A486CBF8B544A</RequestId>
  <HostId>guoping-hangzhou.oss-cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

This error may be caused by the following reasons:

- The body of the response returned by the application server to OSS is not in JSON format, as shown in the following figure:

```
#response to OSS
resp_body = '{"Status":"OK"'
self.send_response(200)
self.send_header('Content-Type', 'application/json')
self.send_header('Content-Length', str(len(resp_body)))
self.end_headers()
self.wfile.write(resp_body)
```

OSS reports the error if `resp_body` is not in valid JSON format. In addition, this error may be caused by other underlying factors, such as the application server returning a stack trace instead of a normal response to OSS because of exceptions.

- The body of the response returned by the application server to OSS carries a BOM in the header.

This problem generally occurs in application servers coded in PHP, which include a BOM header in the response returned to OSS. Therefore, OSS reports the error because three additional bytes (that is, the BOM header) are included

in the response, which does not conform to JSON format. The following figure shows the content included in the packet sent by the application server.

The figure displays a network packet capture for a Hypertext Transfer Protocol (HTTP) response. The packet is 448 bytes on wire and 448 bytes captured. The response is a text/html document. The content of the response is shown in two parts: a hex dump and a text view. The text view shows the following content: `\357\273\277{"Status": "Ok"}`. The hex dump shows the following bytes: `ef bb bf 7b 22 53 74 61 74 75 73 22 3a 22 4f 6b 22 7d`. The `ef bb bf` bytes are highlighted in red, indicating they are the BOM header.

In the preceding figure, the `ef bb bf` bytes are the three additional bytes of the BOM header.



#### Note:

To resolve this issue, remove the BOM header in the response returned by the application server to OSS.

#### • Error status

Error status codes, such as 502 and 400, are errors that are returned due to incorrect callback functions, as shown in the following figure.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>CallbackFailed</Code>
  <Message>Error status : 502.</Message>
  <RequestId>5763F3C8B47FB809EB5700DA</RequestId>
  <HostId>guoping-hangzhou.oss-cn-hangzhou.aliyuncs.com</HostId>
</Error>
```



#### Note:

An error status code, such as 400, 404, or 403, is returned to indicate the HTTP status returned by the application server to OSS. A return of status code 200 indicates the operation is successful.

Error status code 502 is returned when the web service is not enabled on the application server, meaning the server cannot receive the callback request sent by OSS.

- **Timeout**

The following figure shows a timeout error.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>CallbackFailed</Code>
  <Message>Error status : -1.call 10.157.223.86:8080 reply timeout, cost:5000ms, timeout:5000ms (err:-4, errno:115)</Message>
  <RequestId>5763F580727EC3D8022E14B6</RequestId>
  <HostId>guoping-hangzhou.oss-cn-hangzhou.aliyuncs.com</HostId>
</Error>
```



**Note:**

For security reasons, OSS waits to receive the callback response for a maximum of 5 seconds. If the response is not returned, OSS disconnects from the application server and returns a timeout error to the client. The IP address included in the error message can be ignored.

## 7 STS common errors and troubleshooting

STS AssumeRole common errors and causes are listed as follows:

| SN | Error  | Cause  |
|----|--|--|
| 1  | <b>ErrorCode: NoPermission</b><br><b>ErrorMessage: Roles may not be assumed by root accounts.</b>          | AssumeRole is called using the primary account's key. Use the subaccount's key.  |
| 2  | <b>ErrorCode: missingsecuritytoken</b><br><b>ErrorMessage: securitytoken is mandatory for this action.</b> | AssumeRole is called using the temporary account's key. Use the subaccount's key.  |
| 3  | <b>Error code: InvalidAccessKeyId.NotFound</b><br><b>Error message: Specified access key is not found</b>  | The AccessKeyId is invalid. Check whether it is entered correctly. No spaces are left at both sides of AccessKeyId.  |
| 4  | <b>Error code: InvalidAccessKeyId.Inactive</b><br><b>Error message: Specified access key is disabled.</b>  | The subaccount's key used is disabled. Enable or replace the key. You can navigate to Resource Access Management > User Management > Management > User Details > User AccessKey on the console to check whether the key is disabled and enable it. |

| SN | Error   | Cause   |
|----|---|---|
| 5  | <b>ErrorCode:</b> InvalidParameter.PolicyGrammar<br><b>ErrorMessage:</b> The parameter Policy has not passed grammar check. | <p>The authorization policy specified during role play is invalid. You may specify or not specify an authorization policy for AssumeRole. If an authorization policy is specified, the permission for a temporary account is a combination of the specified authorization policy and the permissions for the role. If no authorization policy is specified, the permission for a temporary account is the permissions for the role. When this error is reported, check the specified authorization policy. It is not recommended that an authorization policy is specified when a temporary account assumes a role. If you do need a authorization policy , use the <a href="#">RAM Policy Editor</a> to generate it.</p> |

| SN | Error  | Cause   |
|----|--|---|
| 6  | <p><b>ErrorCode:</b> InvalidParameter.RoleSessionNameErrorMessage: The parameter RoleSessionName is wrongly formed.</p>                | <p>The RoleSessionName specified for AssumeRole is invalid. This parameter is used to identify different tokens to indicate who is using a specific token, which facilitates audit. Format: <code>^[a-zA-Z0-9.@- _]+\$</code>. The parameter has a length of 2 to 32 characters.</p> <p>For more information, see <a href="#">Role assuming operation interface</a>. For example, the names like <code>a</code>, <code>1</code>, <code>abc</code>, <code>* abc</code>, and <code>Teenage Mutant Ninja Turtles</code> are invalid.</p> |
| 7  | <p><b>ErrorCode:</b> InvalidParameter.DurationSeconds<br/><b>Error message:</b> The Min/Max value of DurationSeconds is 15min/1hr.</p> | <p>When the role is assumed, the specified expiration time is invalid. In other words, the parameter <code>AssumeRoleRequest.setDurationSeconds</code> is invalid. When the role is assumed, the expiration time in seconds can be specified. The valid duration time is between 900 and 3600 seconds. For example, <code>assumeRoleRequest.setDurationSeconds ( 60L * 20 )</code> means that it is valid within 20 minutes.</p>  |

| SN              | Error   | Cause  |
|-----------------|---|--|
| 8               | <p><b>ErrorCode: NoPermissionErrorMessage: No permission perform sts:AssumeRole on this Role. Maybe you are not authorized to perform sts:AssumeRole or the specified role does not trust you</b></p> | <ul style="list-style-type: none"> <li>· <b>Cause 1:</b> The subaccount of AssumeRole has no permissions. You must grant the subaccount an AliyunSTSAssumeRoleAccess system authorization policy. You must navigate to Resource Access Management &gt; User Management &gt; Authorization &gt; Optional Authorization Policy Name to grant the subaccount an AliyunSTSAssumeRoleAccess permission.</li> <li>· <b>Cause 2:</b> The account ID for the subaccount sending a request for assuming a role does not match the “Trusted Account ID” for the role. The role creator needs to confirm and modify the account ID. The account ID for the subaccount is the ID of the primary user who has created this subaccount. The account ID for the role is the account ID for the primary user who has created this role. You must navigate to Resource Access Management &gt; Role Management &gt; Management &gt; Role Details &gt; Editing Basic Information to confirm and modify it.</li> </ul> |
| Issue: 20190816 |   | <ul style="list-style-type: none"> <li>· <b>Cause 3:</b> The role type is incorrect. If the type of roles is classified into “User Role”</li> </ul>  |

**Note:**

- For examples of assuming the Java role, see [GitHub](#).
- For examples of assuming role due to other causes, see [STS SDK user guide](#).

## 8 OSS CORS

---

For Cross Origin Resource Sharing (CORS) introduction and configuration, see CORS best practices.

### Configuration items

CORS configurations have the following items:

- **Origin (AllowedOrigin)**

Allowed origins for CORS request. Multiple origins can be specified at the same time. Complete domain information like `http://10.100.100.100:8001` or `https://www.aliyun.com` must be entered when this parameter is configured. It must be noted that the protocol name `http` or `https` must not be omitted. If the port is not `80` by default, the port must also be configured. If the domain name cannot be determined, you can activate the debugging function of the browser to check the `Origin` in the header. The wildcard `*` can be used in the domain name and only one `*` can be used in each domain name, for example, `https://*.aliyun.com`. If `*` is specified as the origin, cross-domain requests of all origins are allowed.

- **Method**

Select the allowed methods as required. You can select all of them for the debugging process.

- **Allow Header**

The allowed cross-origin request header. Multiple match rules can be configured and must be separated with a carriage return. Each header specified by `Access-Control-Request-Headers` must match a value in `Allowed Header`. Header is readily missing. Unless otherwise specifically required, we recommend it is configured as `*` indicating that all of headers are allowed. The header is not case-sensitive.

- **Expose Header**

List of headers exposed to the browser, that is, the response headers allowing users to access from an application (for example, a Javascript XMLHttpRequest object). No wildcard is allowed. The specific configuration depends on the demands of the application. Only expose the required header. If you do not need to expose this

information, you can leave this field blank. The header is not case-sensitive. This is an optional item.

- Cache time (MaxAgeSeconds)

The cache time for the results returned from browser prefetch requests (OPTIONS requests) for a specific resource. The unit is second. Normally, a relatively large value can be set for the cache time, for example, 60s. This is an optional item.

Generally, the CORS configuration method sets individual rules for each origin that may access the service. If possible, do not include multiple origins in a single rule, and avoid overlap or conflict among multiple rules. For other options, you need only to grant the required permissions.

## Troubleshooting

- Error reporting

CORS configuration errors are reported as follows:

- The browser reports the following errors:

```
OPTIONS http://bucket.oss-cn-beijing.aliyuncs.com/XMLHttpRequest cannot load http://bucket.oss-cn-beijing.aliyuncs.com/. Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin '{yourwebsite}' is therefore not allowed access. The response had HTTP status code 403.
```

- The OSS reports the following errors:

```
< Code > AccessForbidden </ Code >
< Message > CORSResponse: This CORS request is not allowed. This is usually because the evaluation of Origin, request method / Access-Control-Request-Method or Access-Control-Request-Headers are not whitelisted by the resource's CORS spec.</ Message >
```



### Note:

- CORS errors are generally caused by site applications. You can view the request details on the browser. Taking Chrome as an example, you can press **F12** to open **Developer Tool** and then view corresponding elements on **Network**.

- Errors returned from OSS can be obtained through packet capture. If Wireshark is used, you can specify `host host bucket - name . oss - cn - beijing . aliyuncs . com` as the filter.
- Errors returned from OSS can also be obtained through the CORS debugging program [oss-h5-upload-js-direct](#).

For other errors, see [OSS 403 errors and troubleshooting](#).

#### • Troubleshooting

Possible CORS errors include:

- Origin (AllowedOrigin) is set incorrectly.
- Method (AllowedMethod) is set incorrectly.
- Allow Header is set incorrectly.
- Expose Header is set incorrectly.

Debugging procedures:

- Set Origin (AllowedOrigin) to `*` and confirm that this configuration item is correct. If upload is successful after this parameter is set to `*`, it means that Origin (AllowedOrigin) has been configured incorrectly and therefore needs to be checked carefully according to rules.
- Select all options (GET, PUT, DELETE, POST, and HEAD) of Method (AllowedMethod) and confirm that this configuration item is correct.
- Set Allow Header to `*` and confirm that this configuration item is correct.
- Set Expose Header to a specified value or leave the field blank, and confirm that this configuration item is correct.



Note:

On the OSS console, select “Bucket” and configure the aforementioned items by navigation to `Bucket Attribute > Cross Origin Setting`.

## 9 OSS 403

---

Error: UserDisable.UserDisable

If the following error UserDisable is reported when you access OSS:

```
< Code > UserDisable </ Code >
      < Message > userdisable </ message >
```

The error may be caused by two possible reasons:

- Access is denied due to account outstanding payment.

You can click [Billing Management](#) on the [OSS console](#) to check whether an outstanding payment is made. If any, recharge the account in time.



Note:

- Even if an outstanding payment is made, you still can use OSS for 24 hours and your access will be banned later.
- Your historical data is kept for 15 days and will be deleted later.
- Once you see an “Alibaba Cloud OSS Arrearage Message” in the Message Center, recharge your account in time. If not, you cannot use OSS.

- Access is denied due to security reasons.

Click [Notice](#) on the console to enter the Message Center and check the notice of violation on the Security message on the right side. Violation may be caused by various reasons.



Note:

If your account is banned, you must do whatever necessary to recover the use of your account. A new account does not guarantee your normal use.

Error: RequestTimeTooSkewed.The difference between...

If the following error RequestTimeTooSkewed is reported when you access OSS:

```
< Code > RequestTimeTooSkewed </ Code >
< Message > The difference between the request time and
the current time is too large.</ Message >
```

The cause is that the interval between your request time and the time at which OSS receives your request exceeds 15 minutes. Therefore, OSS considers this request to be

invalid due to security reasons and returns this error. You must check the system time of the device sending the request, and adjust it to a correct time according to the time zone.

You may have the following questions:

- What are the criteria for adjusting the system time of the machine or device sending the request?

The system time adopted by OSS is the Greenwich Mean Time (GMT). Therefore, the system time of your device must be adjusted to GMT or to a time within a time zone corresponding to GMT. GMT is the zone time of zero zone, that is the World Standard Time.

If, for example, the system of your device that accesses OSS is configured with GMT +08:00, the system time must be adjusted to a time that is 8 hours earlier than GMT. The other time can be adjusted similarly. The standard time in China is Beijing Time, that is GMT+08:00. If your system time is located at GMT+08:00, your system time only needs to be adjusted to Beijing Time.

- To check your time zone using the Windows system,

click Control Panel > Clock, Language, and Region > Set Date and Time to open the date and time. The +08:00 in the Time Zone column indicates that your device is located in the time zone GMT+08:00.

- If your system is Linux/Unix,

run the `date -R` command to check the time and the time zone. +0800 is shown in the following figure, which indicates that the system time zone of your device is GMT+08:00.

```
[yubin.byb@rs1b04376.et2sqa /home/yubin.byb]
$date -R
Wed, 16 Mar 2016 14:36:42 +0800
```

- Is there a problem of time synchronization when using OSS across multiple regions like Hangzhou, Singapore, and the United States?

There is certainly no problem. The OSS in each region uses GMT and the system time of your device sending the request is also GMT.

**Error: InvalidAccessKeyId.The OSS Access Key Id...**

If the following error is reported when you access OSS:

```
< Code > InvalidAccessKeyId </ Code >
< Message > The OSS Access Key Id you provided does
not exist in our records .</ Message >
```

The possible cause is that your AccessKeyID is disabled or does not exist. You can troubleshoot the error as follows:

Log on to [AccessKey management](#) on the Alibaba Cloud console to confirm that the AccessKeyID used for accessing OSS does exist and has been activated.

- If your AccessKeyID is disabled, activate it.
- If your AccessKeyID does not exist, create a new AccessKeyID and use it to access OSS.

**Error: AccessDenied.The bucket you are attempting to...**

If the following error is reported when you access OSS:

```
< Code > AccessDenied </ Code >
< Message > The bucket you are attempting to access must
be addressed using the specified endpoint . Please
send all future requests to this endpoint .</ Message >
```

The cause is that the endpoint you use to access the bucket is incorrect. For endpoint details, see [OSS basic concepts](#).

How can we find out a correct endpoint? If the SDK is abnormal as follows or returns the following error:

```
< Error >
  < Code > AccessDenied </ Code >
  < Message > The bucket you are attempting to access
must be addressed using the specified endpoint . Please
send all future requests to this endpoint .</ Message >
  < RequestId > 56EA **** 3EE6 </ RequestId >
  < HostId > my - oss - bucket -****. aliyuncs . com </ HostId >
  < Bucket > my - oss - bucket -****</ Bucket >
  < Endpoint > oss - cn -****. aliyuncs . com </ Endpoint >
</ Error >
```

- Then `oss - cn -****. aliyuncs . com` in the `endpoint` is the correct endpoint. You must use `http://oss - cn -****. aliyuncs . com` or `https://oss - cn -****. aliyuncs . com` as the endpoint to access OSS.
- If the `endpoint` is not shown in the error returned, you must log on to OSS console, and on the Overview page find out the bucket you are attempting to

access. Then click the bucket to enter the Bucket Overview page. On the OSS Domain Name area, you can see the domain names of the intranet and the Internet.

- The Internet domain name is used to access OSS on the Internet. The intranet domain name is used to internally access OSS on the intranet of Alibaba Cloud. For example, if you access OSS on your ECS, you can use an intranet domain name.
- Endpoint is composed of the domain name (excluding the bucket part) and the access protocol. For example, the Internet domain name of OSS in the preceding picture is `oss-****.aliyuncs.com`. Therefore, the Internet endpoint is `http://oss-cn-****.aliyuncs.com` and similarly its intranet endpoint is `http://oss-cn-****-internal.aliyuncs.com`.

Error: ImageDamage.The image file may be damaged

If the following error is reported when you access OSS:

```
< Code > ImageDamage </ Code >
< Message > The image file may be damaged .</ Message >
```

This error indicates that part of the image file message is lost or damaged, and the image cannot be identified or processed. You may have a question that an image can be processed locally by an image processor but the OSS reports an error. The cause is that the image processor does some processing of the damaged image but the OSS service currently does not have this function.

Error: AccessDenied.AccessDenied

If the following error is reported when you access OSS:

```
< Code > AccessDenied </ Code >
< Message > AccessDenied </ Message >
```

This error indicates that the user accessing OSS has no permissions for the current operation. The correct `AccessKeyId / AccessKeySecret` must be used. If the account you are using is a subaccount/temporary account (STS), you must confirm your current permissions.

Confirmation method:

Check your permissions on the [RAM console](#). Click User management and click User who needs to confirm the permission, then click User Authorization Policy and

**Authorization Policy for Group.** Confirm the current account has been granted the permissions to operate on the bucket/object.

**Error: SignatureDoesNotMatch.** The request signature we calculated...

If the following error is reported when you access OSS:

```
< Code > SignatureDoesNotMatch </ Code >
< Message > The request signature we calculated does not
match the signature you provided . Check your key and
signing method .</ Message >
```

**Troubleshoot the error as follows:**

**1. Check the endpoint.**

Check whether there is a bucket before the endpoint, whether there is unnecessary / behind the endpoint, and whether there are unnecessary spaces at two sides of the endpoint. For example, the endpoint `http://my-bucket.oss-cn-hangzhou.aliyuncs.com` and `http://oss-cn-hangzhou.aliyuncs.com/` are invalid, while `http://oss-cn-hangzhou.aliyuncs.com` and `https://oss-cn-hangzhou.aliyuncs.com` are valid domain names.

**2. Check the AccessKeyID/AccessKeySecret.**

Confirm that the AccessKeyID/AccessKeySecret is correct. Make sure there are no spaces at two sides of the AccessKeyID/AccessKeySecret, especially when it is copied and pasted.

**3. Check the BucketName/ObjectKey.**

Make sure that the BucketName/ObjectKey is valid and compliant with the naming rule.

- **Bucket nomenclature:** The name of a bucket only consists of lower-case letters, numbers, and hyphens (-) and must start with a lower-case letter or number. The length must be between 3 bytes and 63 bytes.
- **Object nomenclature:** The name of an object adopts UTF-8 codes with a length of 1 to 1,023 bytes. The name cannot start with “/” or “\”.

**4. If your own signature is used, you must follow the signature method provided by OSS SDK.**

OSS SDK supports URL/Header signatures. For more information, see the SDK documentation.

5. If your environment is not suitable for SDK use but you do need to use your signature, see [User signature verification](#) for the signature method. You must check each signature field carefully.

A visual signature tool is provided on the OSS forum. You must compare each signature field and the final signature. The signature tool is available at the [Signature tool address](#).

6. If you use a proxy, you must check whether the proxy server has been configured with an additional header.

#### Other errors

You must judge the causes based on the error codes and messages returned from the SDK. The error messages indicate the error causes. If you suspect that the error is related with the network environment, you can use [ossutil](#) for error troubleshooting and the ossutil may give possible causes.