

Cloud Object Storage

Troubleshooting



Copyright Notice

©2013–2023 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Trademark Notice

 Tencent Cloud

This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies ("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Troubleshooting

 Obtaining RequestId

 Slow Upload over Public Network

 403 Error for COS Access

 Resource Access Error

 404 NoSuchKey is Returned

 A Video Cannot be Played

 A URL Points to a Wrong File

 “HTTP ERROR 403” Is Returned When I Access COS Using a CDN Domain

 Accessing a Custom Origin Domain over HTTPS Failed

 POST Object Common Exceptions

Troubleshooting

Obtaining RequestId

Last updated: 2023-09-14 17:57:19

Feature Overview

The COS server will generate an ID (`RequestId`) for every request sent to COS. This document describes how to obtain `RequestId` in different scenarios.

Through the Console

1. Log in to the [Object Storage Console](#) and click **Bucket List** in the left navigation bar to enter the bucket list page.
2. Click the name of the target bucket.
3. Press `F12` to open the browser's Developer Tools.
4. Click **Network** at the top of the developer tools.

Name	Status	Type	Initiator	Size	T. Waterfall
event?e=console:cgi+console:cgi+co...	200	ping	setting?type=fi...	1.5 kB	<div style="width: 100%;">[green bar]</div>

5. Click **Download** next to the file you want to download. In the developer tools page, enter the file name to filter, select the file, and click **Headers**. Obtain the `RequestId` information in the **Response Headers** section.

Name	Value
info?_uin=10002...	1

Header	Value
X-Cos-Request-Id	c8857c-4f5... NjRIVzR... KMDfOTU... = ,NjRIVzR... NTZfODR... jYl9mM2QxNj U= ... fMTcynZkMD... M2Q=

From an Unsuccessful Access

When a file access fails, you can obtain the **RequestId** node information from the XML data displayed on the page.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

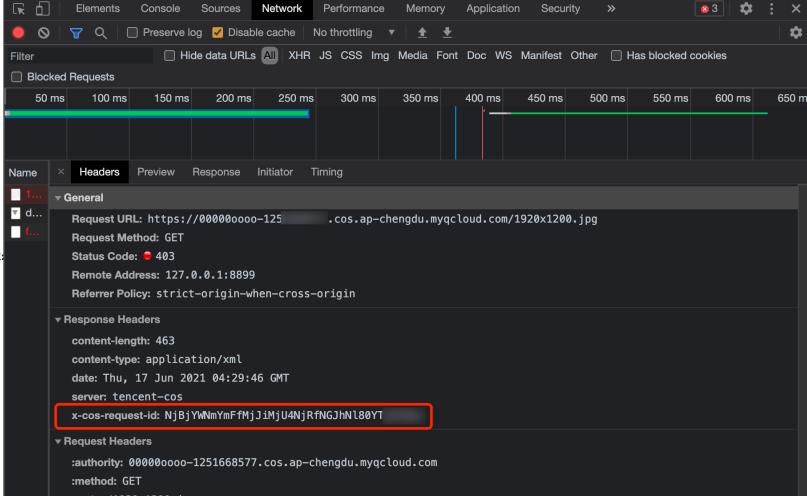
```
<Error>
  <SCRIPT id="allow-copy_script">(function agent() { let unlock = false document.addEventListener('allow_copy', 'contextmenu', 'selectstart', 'mousedown', 'mouseup', 'mousemove', 'keydown', 'keypress', 'keyup', ] const rej (e.stopImmediatePropagation) e.stopImmediatePropagation() } } copyEvents.forEach((evt) => { document.documentElementE
  <Code>AccessDenied</Code>
  <Message>Access Denied.</Message>
  <Resource>....</Resource>
  <RequestId>NjBjYWNmYmFmJiMjU4NjRfNGJhNl80YT</RequestId>
  <TraceId>OGVmYz21MmQzYjA20WnhODk0NTRKMTBi0WVmMDAxODc00WRk2jk02DM1NmI1M2E2MTR1Y2MzZDhmNmI5MWI1OWI2NmQ0YjJkZWE3N
</Error>
```

You can also obtain it as follows:

1. Press **F12** to open the browser's Developer Tools.
2. Click **Network** at the top of the page, select "All" type, and you can find the **RequestId** field information in the Response Headers.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
  <SCRIPT id="allow-copy_script">(function agent() { let unlock = false document.addEventListener('allow_copy', (event) => { unlock = event.detail.unlock }) const copyEvents = [ 'copy', 'cut', 'contextmenu', 'selectstart', 'mousedown', 'mouseup', 'mousemove', 'keydown', 'keypress', 'keyup', ] const rejectOtherHandlers = (e) => { if (unlock) { e.stopPropagation() if (e.stopImmediatePropagation) e.stopImmediatePropagation() } } copyEvents.forEach((evt) => { document.documentElement.addEventListener(evt, rejectOtherHandlers, { capture: true }) }) })()</SCRIPT>
  <Code>AccessDenied</Code>
  <Message>Access Denied.</Message>
  <Resource>00000000-125 .cos.ap-chengdu.myqcloud.com/1920x1200.jpg</Resource>
  <RequestId>NjBjYWNmYmFmJiMjU4NjRfNGJhNl80YT</RequestId>
  <TraceId>OGVmYz21MmQzYjA20WnhODk0NTRKMTBi0WVmMDAxODc00WRk2jk02DM1NmI1M2E2MTR1Y2MzZDhmNmI5MWI1OWI2NmQ0YjJkZWE3N
</Error>
```



Using SDKs

Since there are too many APIs included in the SDK to list them all, the following examples demonstrate how to obtain the **RequestId** for the current operation using the **Upload File API** as an example.

Using the .NET SDK

```
try
{
    string bucket = "examplebucket-1250000000"; //Bucket format: BucketName-APPID
    string cosPath = "test.cs"; // Object key
    byte[] data = System.Text.Encoding.Default.GetBytes("Hello COS"); // Binary data
    PutObjectRequest putObjectRequest = new PutObjectRequest(bucket, cosPath, data);

    PutObjectResult result = cosXml.PutObject(putObjectRequest);
    string requestId = result.responseHeaders.GetValueOrDefault("x-cos-request-id")[0];
    Console.WriteLine(requestId);
}

catch (COSXML.CosException.CosClientException clientEx)
{
    //Request failed
    Console.WriteLine("CosClientException: " + clientEx);
}
catch (COSXML.CosException.CosServerException serverEx)
{
    //Request failed
    Console.WriteLine("CosServerException: " + serverEx.GetInfo());
```

}

Using the Go SDK

```
package main

import (
    "context"
    "fmt"
    "net/http"
    "net/url"
    "strings"
    "github.com/tencentyun/cos-go-sdk-v5"
)

func main() {
    // Replace examplebucket-1250000000 and COS_REGION with the actual information
    u, _ := url.Parse("https://examplebucket-1250000000.cos.COS_REGION.myqcloud.com")
    b := &cos.BaseURL{BucketURL: u}
    c := cos.NewClient(b, &http.Client{
        Transport: &cos.AuthorizationTransport{
            SecretID: "SECRETID",
            SecretKey: "SECRETKEY",
        },
    })
    // The object key (Key) is the unique identifier for an object within a bucket.
    // For example, in the object access domain examplebucket-1250000000.cos.COS_REGION.myqcloud.com/test.go, the object
    key is test.go.
    name := "test.go"
    // 1. Upload the object with a string.
    f := strings.NewReader("Hello COS")

    response, err := c.Object.Put(context.Background(), name, f, nil)
    if err != nil {
        // The error message directly contains the RequestId field.
        panic(err)
    }
    requestId := response.Header.Get("X-Cos-Request-Id")
    fmt.Println(requestId)
}
```

Using the Java SDK

```
// 1. Initialize the user credentials (secretId, secretKey).
String secretId = "SECRETID";
String secretKey = "SECRETKEY";
COSCredentials cred = new BasicCOSCredentials(secretId, secretKey);
// 2. Set the bucket region. For abbreviations of COS regions, please visit
https://cloud.tencent.com/document/product/436/6224.
// clientConfig contains set methods to set region, HTTPS (HTTP by default), timeout, proxy, etc. For details, see the source
code or the Java SDK FAQs.
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// The HTTPS protocol is recommended.
clientConfig.setHttpProtocol(HttpProtocol.https);
// 3. Generate a COS client.
COSClient cosClient = new COSClient(cred, clientConfig);
// Enter the bucket name in the format of BucketName-APPID.
String bucketName = "examplebucket-1250000000";

String content = "Hello COS";
```

```
String key = "test.java";
PutObjectResult putObjectResult = cosClient.putObject(bucketName, key, content);
String requestId = putObjectResult.getRequestId();
System.out.println(requestId);
```

Using the Python SDK

```
# -*- coding=utf-8
from qcloud_cos import CosConfig
from qcloud_cos import CosS3Client
import sys
import os
import logging

# Under normal circumstances, use the INFO log level. To locate issues, change it to DEBUG, and the SDK will print
# communication information with the server.
logging.basicConfig(level=logging.INFO, stream=sys.stdout)

# 1. Set user attributes, including secret_id, secret_key, region, etc. Appid has been removed from CosConfig, please include
# Appid in the Bucket parameter. Bucket is composed of BucketName-Appid.
secret_id = os.environ['COS_SECRET_ID']      # User SecretId. We recommend that you use a sub-account key and follow the
# principle of least privilege to reduce risks. For more information on how to obtain a sub-account key, visit
# https://cloud.tencent.com/document/product/598/37140.
secret_key = os.environ['COS_SECRET_KEY']      # User SecretKey. We recommend that you use a sub-account key and follow
# the principle of least privilege to reduce risks. For more information on how to obtain a sub-account key, visit
# https://cloud.tencent.com/document/product/598/37140.
region = 'ap-beijing'      # Replace it with the actual region, which can be viewed in the console at
# https://console.cloud.tencent.com/cos5/bucket.

        For a list of all regions supported by COS, visit https://cloud.tencent.com/document/product/436/6224
token = None          # Token is required for temporary keys but not permanent keys. For more information about how to
# generate and use a temporary key, see https://cloud.tencent.com/document/product/436/14048.
scheme = 'https'      # Specify whether to use HTTP or HTTPS protocol to access COS. This is optional and is https by
# default.

config = CosConfig(Region=region, SecretId=secret_id, SecretKey=secret_key, Token=token, Scheme=scheme)
client = CosS3Client(config)

try:
    response = client.put_object(
        Bucket='examplebucket-1250000000',
        Key='exampleobject',
        Body=b'abcdefg'
    )

    View request-id in response when the request is successfully returned.
    if 'x-cos-request-id' in response:
        print(response['x-cos-request-id'])

    Viewing request - id through Exception Handling
    except CosServiceError as e:
        print(e.get_request_id())

```

Using the JavaScript SDK

```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* Required */
  Region: 'COS_REGION', /* Required */
  Key: 'test.js', /* Required */
  StorageClass: 'STANDARD',
  Body: 'Hello COS',
  onProgress: function(progressData) {
```

```
        console.log(JSON.stringify(progressData));
    }
}, function(err, data) {
    var requestId = (err || data).headers['x-cos-request-id'];
    console.log(requestId );
});
});
```

Using the Node.js SDK

```
var COS = require('cos-nodejs-sdk-v5');
var cos = new COS({
    SecretId: 'SECRETID',
    SecretKey: 'SECRETKEY'
});

cos.putObject({
    Bucket: 'examplebucket-1250000000', /* Required */
    Region: 'COS_REGION', /* Required */
    Key: 'test.nodejs', /* Required */
    StorageClass: 'STANDARD',
    Body: Buffer.from('Hello COS'),
    onProgress: function(progressData) {
        console.log(JSON.stringify(progressData));
    }
}, function(err, data) {
    var requestId = (err || data).headers['x-cos-request-id'];
    console.log(requestId );
});
});
```

Using the Weixin Mini Program SDK

```
var COS = require('cos-wx-sdk-v5');
var cos = new COS({
    SecretId: 'SECRETID',
    SecretKey: 'SECRETKEY'
});

cos.putObject({
    Bucket: 'examplebucket-1250000000', /* Required */
    Region: 'COS_REGION', /* Required */
    Key: 'test.js', /* Required */
    StorageClass: 'STANDARD',
    Body: 'Hello COS',
    onProgress: function(progressData) {
        console.log(JSON.stringify(progressData));
    }
}, function(err, data) {
    var requestId = (err || data).headers['x-cos-request-id'];
    console.log(requestId );
});
});
```

Using the PHP SDK

```
$secretId = "SECRETID"; // "SecretId of your Tencent Cloud API key";
$secretKey = "SECRETKEY"; // "SecretKey of your Tencent Cloud API key";
$region = "COS_REGION"; // Set the default bucket region
$cosClient = new Qcloud\Cos\Client(
    array(
        'region' => $region,
        'schema' => 'https', // Protocol header, default is http
        'credentials' => array(
            'accessKeyId' => $secretId,
            'accessKeySecret' => $secretKey
        )
    )
);
```

```

'secretId' => $secretId,
'secretKey' => $secretKey));
# Upload a File
## putObject (Upload interface, supports uploading files up to 5GB)
#### Upload a String from Memory
try {
$bucket = "examplebucket-1250000000"; // Bucket name format: BucketName-APPID
$key = "test.php"; // The key here is the object key, which is the unique identifier of the object in the bucket.
$result = $cosClient->putObject(array(
'Bucket' => $bucket,
'Key' => $key,
'Body' => 'Hello COS'));
$requestId = $result['RequestId'];
print_r($requestId);
} catch (\Exception $e) {
echo "$e\n";
}
}

```

Using the iOS SDK

```

QCloudCOSXMLUploadObjectRequest* put = [QCloudCOSXMLUploadObjectRequest new];
/** Path of the local file. Ensure that the URL starts with "file://" in the following format:
1. [NSURL URLWithString:@"file:///var/mobile/Containers/Data/Application/DBPF7490-D5U8-4ABF-A0AF-
CC49D6A60AEB/Documents/exampleobject"]
2. [NSURL fileURLWithPath:@"/var/mobile/Containers/Data/Application/DBPF7490-D5U8-4ABF-A0AF-
CC49D6A60AEB/Documents/exampleobject"]
*/
NSURL* url = [NSURL fileURLWithPath:@"file URL"];
// Bucket name in the format of BucketName-Appid, which can be viewed in the COS console at
// https://console.cloud.tencent.com/cos5/bucket.
put.bucket = @"examplebucket-1250000000";
// Object key, i.e., the full path of a COS object. If the object is in a directory, the path should be "video/xxx/movie.mp4"
put.object = @"exampleobject";
// Content of the object to be uploaded. You can pass variables of the NSData* or NSURL* type.
put.body = url;
// Monitor the upload progress
[put setSendProcessBlock:^(int64_t bytesSent,
                           int64_t totalBytesSent,
                           int64_t totalBytesExpectedToSend) {
    // bytesSent           Number of bytes to be sent in this instance (a large file may require multiple transmissions)
    // totalBytesSent      Total bytes sent
    // totalBytesExpectedToSend  Total number of bytes to be sent in this upload (i.e., the file size)
}];
// Monitor the upload result
[put setFinishBlock:^(QCloudUploadObjectResult *result, NSError *error) {
    // Obtain RequestId
    [result._originHTTPURLResponse.allHeaderFields objectForKey:@"x-cos-request-id"]
}];
[put setInitMultipleUploadFinishBlock:^(QCloudInitiateMultipartUploadResult *
                                       multipleUploadInitResult,
                                       QCloudCOSXMLUploadObjectResumeData resumeData) {
    // After initializing the multipart upload, this callback block will be executed, where you can obtain the resumeData and
    // uploadId.
    NSString* uploadId = multipleUploadInitResult.uploadId;
}];
[[QCloudCOSTransferMangerService defaultCOSTransferManager] UploadObject:put];

```

Using the Android SDK

```

// 1. Initialize TransferService. You should use the same TransferService for the same configuration
TransferConfig transferConfig = new TransferConfig.Builder()

```

```
.build());
CosXmlServiceConfig cosXmlServiceConfig = new CosXmlServiceConfig.Builder()
    .setRegion(COS_REGION)
    .builder();
CosXmlService cosXmlService = new CosXmlService(context, cosXmlServiceConfig, credentialProvider);
TransferService transferService = new TransferService(cosXmlService, transferConfig);

// 2. Initialize PutObjectRequest
String bucket = "examplebucket-1250000000"; // Bucket name in the format: BucketName-APPID
String cosPath = "exampleobject"; // The location identifier of the object in the bucket, i.e. the object key
String srcPath = "examplefilepath"; // Absolute path to the local file
PutObjectRequest putObjectRequest = new PutObjectRequest(bucket,
    cosPath, srcPath);

// 3. Call the upload method to upload the file
final COSUploadTask uploadTask = transferService.upload(putObjectRequest);
uploadTask.setCosXmlResultListener(new CosXmlResultListener() {
    @Override
    public void onSuccess(CosXmlRequest request, CosXmlResult result) {
        // Upload successful, you can obtain the requestId here.
        String requestId = result.getHeader("x-cos-request-id");
    }

    @Override
    public void onFail(CosXmlRequest request,
        CosXmlClientException clientException,
        CosXmlServiceException serviceException) {
        // Only CosXmlServiceException will have requestId
        if (serviceException != null) {
            String requestId = serviceException.getRequestId();
        }
    }
});
```

Slow Upload over Public Network

Last updated: 2023-09-14 18:08:34

Problem

- **Phenomenon 1:**
- The upload speed is slow (below 8 Mbps) over a home network, while it is normal over a corporate network.
- The upload speed is slow (below 8 Mbps) over a corporate network, while it is normal using a 4G network.
- **Phenomenon 2:** The upload speed is slow when using a custom domain.

Possible Cause

- Symptom 1

1. The speed is slow due to your ISP and network condition.
2. CORS is used and thus the speed is slow.

- Symptom 2: CNAME maps the custom domain name to other products (e.g., Content Delivery Network (CDN), Cloud Virtual Machine (CVM), or Anti-DDoS) before mapping to COS.

Solution

- If you encounter [Phenomenon 1](#), you can troubleshoot by checking the client's network environment. For detailed steps, please refer to [Troubleshooting Client Network](#).
- If you encounter [Phenomenon 2](#), you can improve the transmission efficiency by modifying the custom domain name resolution to reduce the transmission relay links. For detailed instructions, please refer to [Modifying Custom Domain Name Resolution](#).

Instructions

Checking the client network

1. Run the following command to check whether the ISP of the IP address is the same as that of the client network:

```
ping COS domain
```

Example:

```
ping examplebucket-1250000000.cos.ap-beijing.mqcloud.com
```

- If yes, skip to [Step 3](#).
- If not, proceed to [step 2](#).

2. Using Google Chrome as an example, check if the browser has a proxy configured.

2.1 Open Google Chrome, click on the  in the top-right corner, and select **Settings** to open the settings page.

2.2 Click **Advanced**, select **Proxy settings for your computer** under the "System" section, and open the operating system's settings window.

Check if a proxy is set.

- If yes, disable the proxy.
- If not, perform [Step 3](#).

3. Check if the Wi-Fi router being used has any speed limitations.

- If yes, allocate the bandwidth as needed.
- If not, please go to [Step 4](#).

4. Check the current network's transfer performance when uploading to COS.

Using the COSCMD tool as an example, test the upload and download performance of a 20MB object.

```
coscmd probe -n 1 -s 20
```

The results will be similar to the following, displaying the average speed (Average), the lowest speed (Min), and the highest speed (Max).

```
D:\work>coscmd probe -n 1 -s 20
Upload tmp_test_20M => cos://[REDACTED]/tmp_test_20M
[success]
Download cos://[REDACTED]/tmp_test_20M => tmp_test_20M
[success]
Success Rate: [1/1]
 20M TEST      Average      Min      Max
 Upload        12. 6MB/s   12. 6MB/s   12. 6MB/s
 Download       10. 2MB/s   10. 2MB/s   10. 2MB/s
```

Note

Please install the COSCMD tool first. After installation, execute commands using the command line tool. For more information, see [COSCMD Tool](#).

5. Browse [Speed Test](#) to test the speed and compare the tested speed with data obtained in [Step 4](#) to determine whether the client has used the maximum bandwidth.
 - If the speeds obtained in Step 4 are slower than the client bandwidth, please [contact us](#).
 - If the speeds obtained in Step 4 equal to the client bandwidth but have not reached the speed promised by your ISP, please contact your ISP.
 - If the speeds obtained in Step 4 equal to the client bandwidth and reached the bandwidth promised by your ISP, please perform [Step 6](#).
6. Check if there are domestic clients accessing overseas node buckets, or overseas clients accessing domestic node buckets.
 - Yes, it is recommended to use COS's global acceleration feature. For more information, please refer to [Tencent Cloud COS Global Acceleration for Faster Access by Global Users](#) and [Efficient Transmission Practices with COS Global Acceleration](#).
 - If not, [contact us](#).

Modifying DNS record for your custom domain

1. Check whether the custom domain is mapped to a COS domain.

- Yes, please [contact us](#).

Common COS domain names are as follows:

```
XXX.cos.ap-beijing.myqcloud.com (default COS domain)
XXX.cos.accelerate.myqcloud.com (COS global acceleration domain)
XXX.cos-website.ap-beijing.myqcloud.com (COS static website domain)
XXX.picbj.myqcloud.com (COS's default CI domain)
```

- No, please proceed to [Step 2](#).

Common non-COS domain names are as follows:

```
XXX.file.myqcloud.com or XXX.cdn.dnsv1.com (Tencent Cloud's CDN default domain)
XXX.w.kunlungr.com (aliyunCDN default domain)
```

2. Resolve the custom domain name's CNAME to the required COS domain name and proceed with data upload.

For example, `upload.mydomain.com cname XXX.cos.ap-beijing.myqcloud.com`. For detailed instructions, please refer to [Enabling Custom Origin Domain Name](#).

3. Modify the default COS domain name in the client.

For example, using C# code:

```
CosXmlConfig config = new CosXmlConfig.Builder()
    .SetConnectionTimeoutMs(60000) // Set the connection timeout in milliseconds. Defaults to 45000.
```

```
.SetReadWriteTimeoutMs(40000) // Set the read/write timeout in milliseconds. Defaults to 45000.  
.IsHttps(true) // Set HTTPS as the default request method.  
.SetAppid(appid) // Set the APPID of your Tencent Cloud account  
.SetRegion(region) // Set the default bucket region.  
.SetHost("XXXXXX.com") // Enter the custom domain.  
.SetDebugLog(true) .Build(); // Create the CosXmlConfig object.
```

For more information about how to call other SDKs, see [SDK Overview](#).

403 Error for COS Access

Last updated: 2023-09-14 18:16:23

Problems

- 403 is returned when you use the APIs or SDKs of COS to upload/download files.
- Error code 403 is returned when you access COS resources by using a sub-account or temporary key.
- 403 is returned when you modify the bucket configuration.

Analysis

When the error code 403 appears in the COS request, you can refer to the following procedures to analyze the cause of the problem:

1. Check whether the request is a CORS request. For failed cross-origin requests, COS will return "[AccessForbidden](#)".
2. Check whether the request hits the bucket's referer hotlink protection configuration. For the error code 403 caused by the hotlink protection rule, COS will return "[You are denied by bucket referer rule](#)".
3. Check whether the request is anonymous. For unsigned requests to a non-publicly readable object, COS will return "Access Denied." To set a bucket or object publicly readable, see [Setting Access Permission](#) or [Setting Object Access Permission](#).
4. Check whether the request key and signature are correct.
 - 4.1 COS returns "[InvalidAccessKeyId](#)" when the SecretId is not used correctly in the signature.
 - 4.2 If the local system time is inaccurate, or the request time is after the validity period of the signature, COS will return "[RequestTimeTooSkewed](#)" or "[Request has expired](#)".
 - 4.3 If there is a problem with the calculation method of the signature, COS will return "[SignatureDoesNotMatch](#)".
5. Check whether the sub-account or temporary key that initiated the request has been granted corresponding access permission.
 - 5.1 Verify the sub-account's access permissions. A sub-account without granted resource access permissions (including incorrect resource description elements, such as an erroneous APPID) will return "[Access Denied](#)".
 - 5.2 For requests initiated with a temporary key, the policy entered when the temporary key was applied for limits the scope of accessible resources. For more information, see [Generating and Using Temporary Keys](#).
6. Check whether the requested object is stored in ARCHIVE or DEEP ARCHIVE storage class. For such requests, COS will return "[InvalidObjectState](#)".

Locating and troubleshooting the problem

Message: "Access Denied."

If the following message is displayed when you access COS:

```
<Code>AccessDenied</Code>
<Message>Access Denied.</Message>
```

You can:

1. Log in to the [COS Console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. In the left sidebar, select **Permission Management > Bucket Access Permissions** to enter the bucket access permission management page.
5. In the **Bucket ACL (Access Control List)** area, check whether the COS account is granted access permission.
 - If yes, proceed with the next step.
 - If not, click **Add User** to set the required permissions for the account accessing COS.

Bucket ACL (Access Control List) After authentication, the object can be accessed. You can browse through [Setting Access Permissions](#) or [Authorize users](#).

User ACL

User Type	Account ID	Permission	Operation
Root account	28 33 22	Full control	...

Root account Reads Write Read ACL Write ACL Full control

[Add User](#)

6. Verify whether the account has the granted permission.

- If yes, proceed with the next step.
- If not, click **Edit** to configure again.

7. In the **Permission Policy Settings** area, check whether a policy has been configured for the account.

⚠ Note

- If the bucket is set to **Private Read/Write** but anonymous access is set in the policy, the permission set in the policy takes effect.
- If an action is set to **Allow** and **Deny** in different policies associated with the same sub-account, the action will be denied.
- Policies set for **Everyone** have lower priorities than those set for a **Specified user**.

- If yes, proceed with the next step.
- Please click **Add Policy** and configure the required permissions based on the actual account access during the signing process.

Add Policy

Template 2 Configure Policy

When dealing with authorizations, you should strictly conform to principles of least privilege. You can authorize the user to perform restricted operations (such as only authorize read operations) and access only the resources with specified prefix, to avoid data security risks due to excessive permissions and operations that you don't mean to authorize.

Policy ID

Effect Allow Deny

User

Resource The whole bucket Specified path

Operation All Actions

Condition

[Previous](#) [Finish](#)

8. Verify whether the account has the granted permission.

- If yes, proceed with the next step.
- If not, click **Edit** to configure again.

9. Check whether the value of `q-ak` (case-sensitive) is the `secretID` and `secretKey` of the root account of the bucket.

- If yes, proceed with the next step.
- If not, change the value of `q-ak` to the `secretID` and `secretKey` of the root account.

10. Check whether the resource is accessed by a cross-account.

- If yes, authorize the sub-account by referring to [Authorizing Cross-Account's Sub-account Read/Write Access to Specified File](#).
- If no, proceed to the next step.

11. Verify if a temporary key is being used.

- Yes, verify if the action and allowPrefix provided when applying for the temporary key are correct.
 - For example, when calling `cos.putObject()`, if `name/cos:PutObject` is not specified in the action, it results in a 403 error due to the lack of `putObject` permission.
 - For instance, if the operation key is `1.jpg`, but the `allowPrefix` is set to `test/*` (only allowing operations on the `test/*` path), this means there is no permission for the corresponding path, resulting in a 403 error.
 - For example, the authorized resource scope is for user APPID 1250000000, but the resource description element (resource) is filled with 1250000001, resulting in a 403 error due to the lack of corresponding APPID operation permissions.

Note

Different STS SDKs in various languages use different fields for action and allowPrefix, such as allowActions and allowPrefixes in the STS Java SDK. Please pay attention to the examples in the STS SDK.

- If not, [contact us](#).

Message: “AccessForbidden”

If the following message is displayed when you access COS:

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

You can:

1. Log in to the [COS Console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. In the left sidebar, select **Security Management > CORS Settings** to enter the CORS configuration page.
5. In the "CORS Settings" section, check if it is a cross-origin request.

Origin	Allow-Methods	Allow-Headers	Expose-Headers	Max-age	Vary	Operation
http://www.***.com:8080	PUT	*	ETag Content-Length x-cos-request-id	600	Enabled	Modify Delete

- If yes, proceed with the next step.
- If not, modify the CORS rule.

6. Run the following command to check whether the CORS request is correctly configured:

If the information below is returned, the configuration is correct.

Message: “You are denied by bucket referer rule”

If the following message is displayed when you access COS:

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

You can:

1. Log in to the [COS Console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. In the left sidebar, choose **Security Management > Hotlink Protection Settings** to access the Hotlink Protection Settings page.
5. In "Hotlink Protection", check if hotlink protection is enabled.

Note: COS provides Hotlink Protection to prevent malicious programs from using resource URLs to steal your public network traffic.
For more information, please see [Hotlink Protection Guide](#)

If yes, proceed with the next step.

If not, [contact us](#).

6. Run the following command to check whether hotlink protection is correctly configured:

```
curl 'http://bucket-appid.cos.ap-guangzhou.myqcloud.com/object' -voa /dev/null -H 'referer: the value of referer'
```

If the information below is returned, the configuration is correct.

```
[root@TencentCloud-22-2 centos]# curl 'http://[REDACTED].cos.ap-guangzhou.myqcloud.com/1.gif' -voa /dev/null -H 'referer: http://[REDACTED]' 
* Total: 0 % Received: 0 % Xferd: Average Speed: 0 Time: 0 Time: 0 Time: 0 Current: 0
* Dload: 0 Upload: 0 Total: 0 Spent: 0 Left: Speed: 0* Trying [REDACTED]:80...
* Connected to [REDACTED] ([REDACTED].cos.ap-guangzhou.myqcloud.com ([REDACTED])) port 80 (#0)
> User-Agent: curl/7.7.9
> Accept: */*
> Referer: http://[REDACTED]
< Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: image/gif
< Content-Length: 307293
< Connection: keep-alive
< Date: Tue, 30 Mar 2021 03:15:40 GMT
< ETag: "2cf137c0d3b2a6d1a0a3c1d373fd2a9d"
< Server: tencent-cos
< x-cos-hash-crc64ecma: 63978585775965391
< x-cos-request-id: N1A2Hk2Z0Hf7zhtsN1
< x-cos-version-id: fTgD0UxJgxOT...
[ data not shown]
100 300K 108 300K 0 0 520k 0 ::::: ::::: ::::: 519k
* Connection #0 to host [REDACTED].cos.ap-guangzhou.myqcloud.com left intact
* Closing connection <1>
```

Message: “InvalidAccessKeyId”

If the following message is displayed when you access COS:

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

You can:

1. Check whether the `q-ak` value of `Authorization` in the request signature is correct.
 - If yes, proceed with the next step.
 - If not, modify the value of `q-ak` (case-sensitive), which should be the same as `SecretId` of the key.
2. Go to [API Key Management](#) and check if the API key is enabled.
 - If yes, [contact us](#).
 - If not, enable the API key.

Message: “InvalidObjectState”

If the following message is displayed when you access COS:

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

You can:

Check whether the requested object is stored in ARCHIVE or DEEP ARCHIVE.

- If yes, restore the object first. For more information, please see [POST Object restore](#).
- If not, [contact us](#).

Message: “RequestTimeTooSkewed”

If the following message is displayed when you access COS:

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

You can:

1. Check the client-side time according to the OS as follows:

- Windows System (using Windows Server 2012 as an example):  > Control Panel > Clock, Language, and Region > Set Date and Time.
- Linux system: Run the `date -R` command.

```
(base) [root@VM_32_5_centos ~]# date -R
Tue, 30 Mar 2021 11:44:00 +0800
```

2. Check whether the clock skew between the client and server is larger than 15 minutes.

- If yes, sync the clock.
- If not, [contact us](#).

Message: “Request has expired”

If the following message is displayed when you access COS:

```
<Code>AccessDenied</Code>
<Message>Request has expired</Message>
```

The possible causes are as follows:

- The signature has expired when you initiate the request.
- Your local system time is out of sync with the local time in your time zone.

You need to set the effective time of your signature again or sync the local system time. If the fault persists, [contact us](#).

Message: “SignatureDoesNotMatch”

If the following message is displayed when you access COS:

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

You can:

Check whether the signature calculated on the client side is the same as that calculated on the server side.

- If yes, [contact us](#).
- No, please refer to the [Request Signature](#) documentation and use the [COS Signature Tool](#) to verify your custom signature implementation.

Resource Access Error

404 NoSuchKey is Returned

Last updated: 2023-09-14 18:16:59

Problem

When attempting to access a file, the system indicates that the file cannot be found or is not displayed, returning a 404 NoSuchKey error code.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
  <Code>NoSuchKey</Code>
  <Message>The specified key does not exist.</Message>
  <Resource>[REDACTED]cos.ap-guangzhou.myqcloud.com/3.gif</Resource>
  <RequestId>NjA4MjM4NjRfM[REDACTED]</RequestId>
  <TraceId>OGVmYzZiMmzYjA20WNh0Dk0NTRkMTBi0WVmMDAx0Dc00WRkZjk0ZDM1NmI1M2E2MTR1Y2Mz2DhmNmI5[REDACTED]</TraceId>
</Error>
```

Possible Cause

- The file URL is incorrect.
- The case of the URL is incorrect.

Instructions

Check whether the URL and the case of the accessed file are correct.

- If yes, [contact us](#).
- If not, modify the file URL. For the naming conventions of objects, see [Object Overview](#).

A Video Cannot be Played

Last updated: 2023-09-15 10:14:07

Problem

The uploaded video cannot be played normally (for example, the video has sound only, the video has no sound, or the video cannot be played at all).

Possible Cause

- The codec of the original video is unsupported.
- The original video is corrupted. As a result, the uploaded video cannot be played normally.

Instructions

1. Check whether the codec of the original video is supported.
 - If yes, proceed with the next step.
 - If not, upload a video with a supported codec. For more information, see [Specifications and Limits](#).
2. Check whether the original video can be played normally using VLC or other professional players.
 - If yes, [contact us](#).
 - If not, you are advised to try using Tencent Cloud's transcoding service to fix the video. For more information, see [Video Transcoding](#).

A URL Points to a Wrong File

Last updated: 2023-09-14 18:19:03

Problem

- **Symptom 1:** A single URL points to different files.
- **Symptom 2:** After I updated a file, I still accessed the old version of it.

Possible Cause

- The Content Delivery Network (CDN) cache has not expired.
- The browser did not disable cache.
- The accessed file is hijacked. As a result, the accessed file content is not as expected.

Instructions

Check whether the CDN cache has expired

Check whether the CDN cache has expired as instructed in [Cache Configuration FAQs](#).

- If yes, [check whether your browser has disable cache](#).
- If not, purge CDN URLs or directories by referring to [Purge Cache](#).

Check whether your browser has disabled cache

Note

The following operations use Google Chrome as an example.

1. Open Chrome.
2. Press **F12** to open the Developer tools.
3. Select the **Network** tab and check if **Disable cache** is selected.



- If yes, [check whether the file is hijacked](#).
- If not, select **Disable cache** and restart the browser.

Check whether the file is hijacked

If the file accessed is not as expected (for example, the `content-length` or response headers are not as expected), your file has been hijacked. In this case, you are advised to access it over HTTPS.

“HTTP ERROR 403” Is Returned When I Access COS Using a CDN Domain

Last updated: 2023-09-14 18:21:44

Problem

When accessing Cloud Object Storage (COS) using a Content Delivery Network (CDN) domain, an HTTP ERROR 403 code is returned.

Possible Cause

The CDN acceleration domain is disabled.

Instructions

1. Log in to the [COS Console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. In the left sidebar, select **Domain & Transfer Management > Default CDN Acceleration Domain** to access the Default CDN Acceleration Domain page.
5. In the "Default CDN Acceleration Domain" column, check if the current status is set to disabled.
 - Yes, please [enable the default CDN acceleration domain](#).
 - If no, proceed to the next step.
6. In the **Custom CDN Acceleration Domain** area, check whether the **Status** is **Online**.
 - If yes, [contact us](#).
 - If not, [enable custom acceleration domain](#).

Accessing a Custom Origin Domain over HTTPS Failed

Last updated: 2023-09-14 19:35:25

Problem

When I access a custom origin domain over HTTPS, an error is reported.

Possible Cause

The certificate configuration is incorrect or the custom origin domain is not configured.

Instructions

Using a CDN certificate

1. Log in to the [CDN console](#).
2. On the left sidebar, click **Domain Management**.
3. Click the domain to configure. Then, select the **HTTPS Configuration** tab.
4. In the "HTTPS Configuration" column, click **Go to Configuration** to set up the certificate. For more information, please refer to the [Certificate Configuration](#) documentation for Content Delivery Network.
5. Wait for about 5 minutes. When the CDN domain is deployed again, you can use HTTPS.

Using COS Certificate

1. Log in to the [COS Console](#).
2. Click **Bucket List** on the left sidebar.
3. Click the name of the target bucket to go to the configuration page.
4. In the left sidebar, select **Domain & Transfer Management > Custom Endpoint** to access the Custom Origin Domain page.

Custom Endpoint						
Domain Name	Origin Server Type	CNAME	HTTPS certificate	Status	Operation	
                                                               <img alt="Cloud icon" data-bbox="5070						

POST Object Common Exceptions

Last updated: 2023-09-14 19:40:04

Problems

When POST requests are initiated via COS APIs, the following error messages are returned:

- Condition key q-ak doesn't match the value XXXXXX
- You post object request has been expired, expiration time: 1621188104 but the time now : 1621245817
- The Signature you specified is invalid.
- You must provide condition if you specify a policy in post object request.
- Condition key bucket doesn't match the value [bucket-appid]
- Condition key key doesn't match the value XXXXX
- The body of your POST request is not well-formed multipart/form-data.

Locating and troubleshooting the problem

Error message "Condition key q-ak doesn't match the value XXXXXX"

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>AccessDenied</Code>
<Message>Condition key q-ak doesn't match the value XXXXXX</Message>
```

Possible Cause

q-ak parameter input error.

Solution

- Log in to the Access Management Console and navigate to the [API Key Management](#) page to view the key information.
- Check whether the q-ak parameter is incorrectly set based on the key information:
 - Yes: change the value of the q-ak parameter to the correct SecretId.
 - If not, [contact us](#).

Error message "You post object request has been expired, expiration time: 1621188104 but the time now : 1621245817"

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>AccessDenied</Code>
<Message>You post object request has been expired, expiration time: 1621188104 but the time now :
1621245817</Message>
```

Possible Cause

The value of expiration in the policy has expired.

Solution

Change the value of expiration in the policy.

Note

The expiration value should be later than the current time. It is recommended to set it to the current time plus 30 minutes (UTC time).

Error message "The Signature you specified is invalid."

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>SignatureDoesNotMatch</Code>
<Message>The Signature you specified is invalid.</Message>
```

Possible Cause

Signature calculation error.

Solution

Check whether the POST signature string generation rule is correct by referring to [Request Signature](#):

- If yes, [contact us](#).
- No: use the [COS Signature Tool](#) to calculate the POST request signature again.

Error message "You must provide condition if you specify a policy in post object request."

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>InvalidPolicyDocument</Code>
<Message>You must provide condition if you specify a policy in post object request.</Message>
```

Possible Cause

Policy format error.

Solution

Change the policy format to the standard JSON format by referring to [POST Object](#).

Error message "Condition key bucket doesn't match the value [bucket-appid]"

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>AccessDenied</Code>
<Message>Condition key bucket doesn't match the value [bucket-appid]</Message>
```

Possible Cause

The bucket in the policy is inconsistent with that in the request.

Solution

Use the bucket specified in the policy to initiate the request.

Error message "Condition key key doesn't match the value XXXXX"

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>AccessDenied</Code>
<Message>Condition key key doesn't match the value XXXXX</Message>
```

Possible Cause

The uploaded content does not comply with the policy.

Solution

Upload content that meets the conditions specified in the policy.

Error message "The body of your POST request is not well-formed multipart/form-data."

When POST requests are initiated via COS APIs, the following information is displayed:

```
<Code>MalformedPOSTRequest</Code>
```

◀Message> The body of your POST request is not well-formed multipart/form-data.◀/Message>

Possible Cause

The POST body format is invalid.

Solution

Correct the body format by referring to [POST Object](#).