

对象存储
故障处理
产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

故障处理

- 获取 RequestId 操作指引

- 通过外网上传文件至 COS 缓慢

- 访问 COS 时返回403错误码

- 资源访问异常

 - 访问文件返回404 NoSuchKey

 - 访问视频时无法正常播放

 - 同一个链接访问的文件异常

 - 使用 CDN 域名访问 COS 返回 HTTP ERROR 403

 - 使用 HTTPS 协议访问自定义源站域名报错

- POST Object 常见异常

故障处理

获取 RequestId 操作指引

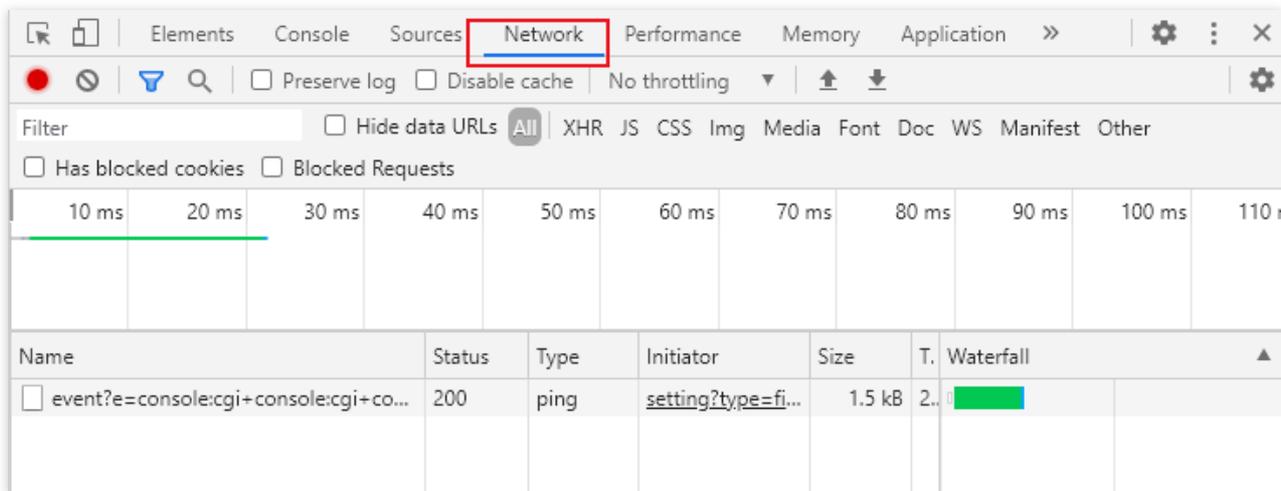
最近更新时间：2024-01-06 11:27:57

简介

在对象存储（Cloud Object Storage，COS）服务中，每次发送请求时，COS 服务端都会为请求生成一个 ID，即 RequestId。本文主要介绍在不同场景下获取 RequestId 的方法。

在控制台通过浏览器获取

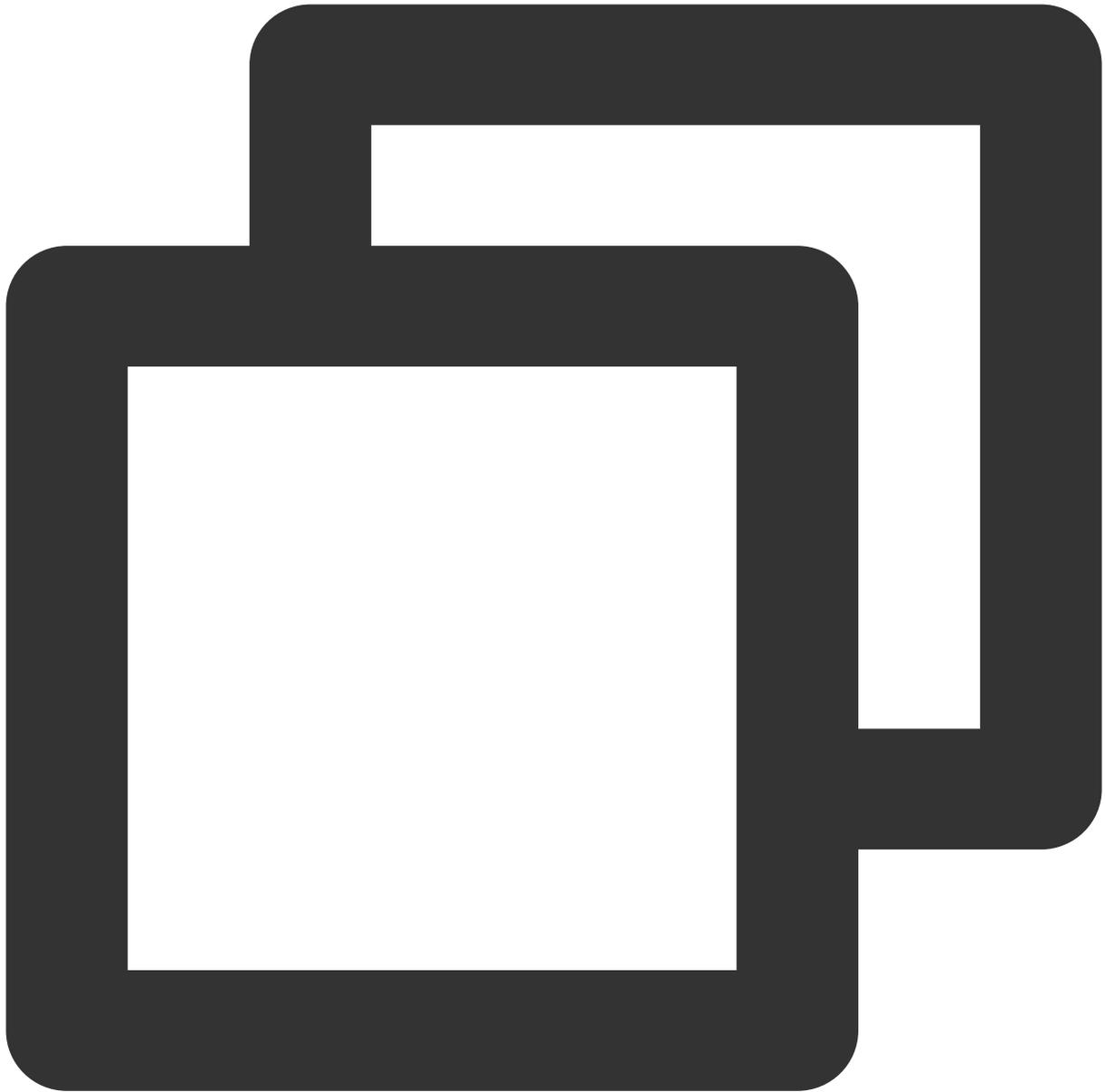
1. 登录 [对象存储控制台](#)，在左侧导航栏中单击**存储桶列表**，进入存储桶列表页。
2. 单击并进入想要访问的存储桶。
3. 按 **F12** 键，进入浏览器的开发者工具页面。
4. 单击开发者工具上方的**Network**。



5. 在需要下载的文件名右侧，单击**下载**，并在开发者工具页面中输入要下载的文件名进行过滤，选择文件，单击**Headers**，在 **Response Headers** 区域中获取 RequestId 信息。

访问文件失败时获取

您可以在访问文件失败时，从页面展示返回的 XML 信息中获取 **RequestId 节点信息**。

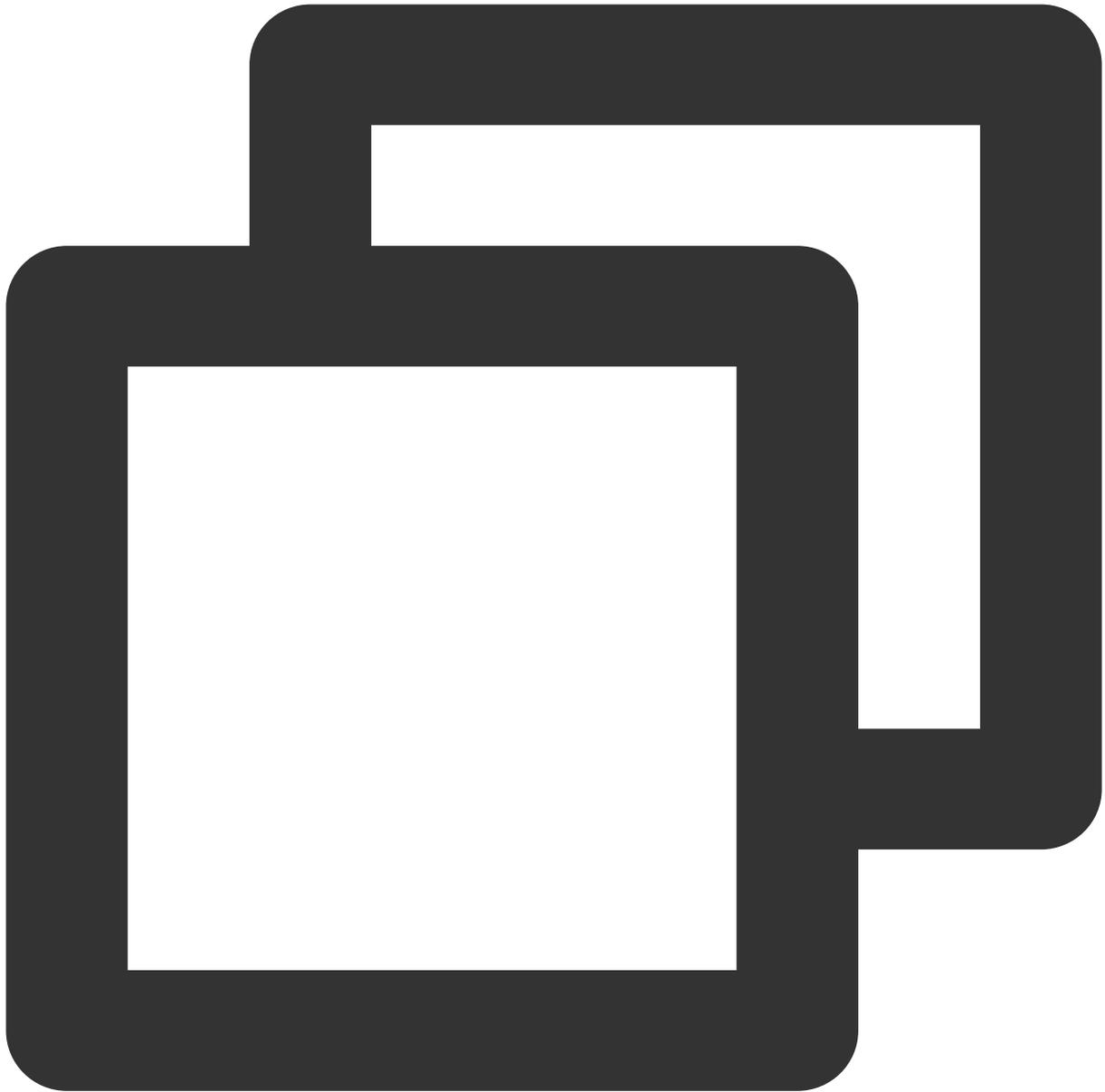


```
try
{
    string bucket = "examplebucket-1250000000"; //存储桶, 格式: BucketName-APPID
    string cosPath = "test.cs"; // 对象键
    byte[] data = System.Text.Encoding.Default.GetBytes("Hello COS"); // 二进制数据
    PutObjectRequest putObjectRequest = new PutObjectRequest(bucket, cosPath, data);

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

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

通过 Go SDK 获取



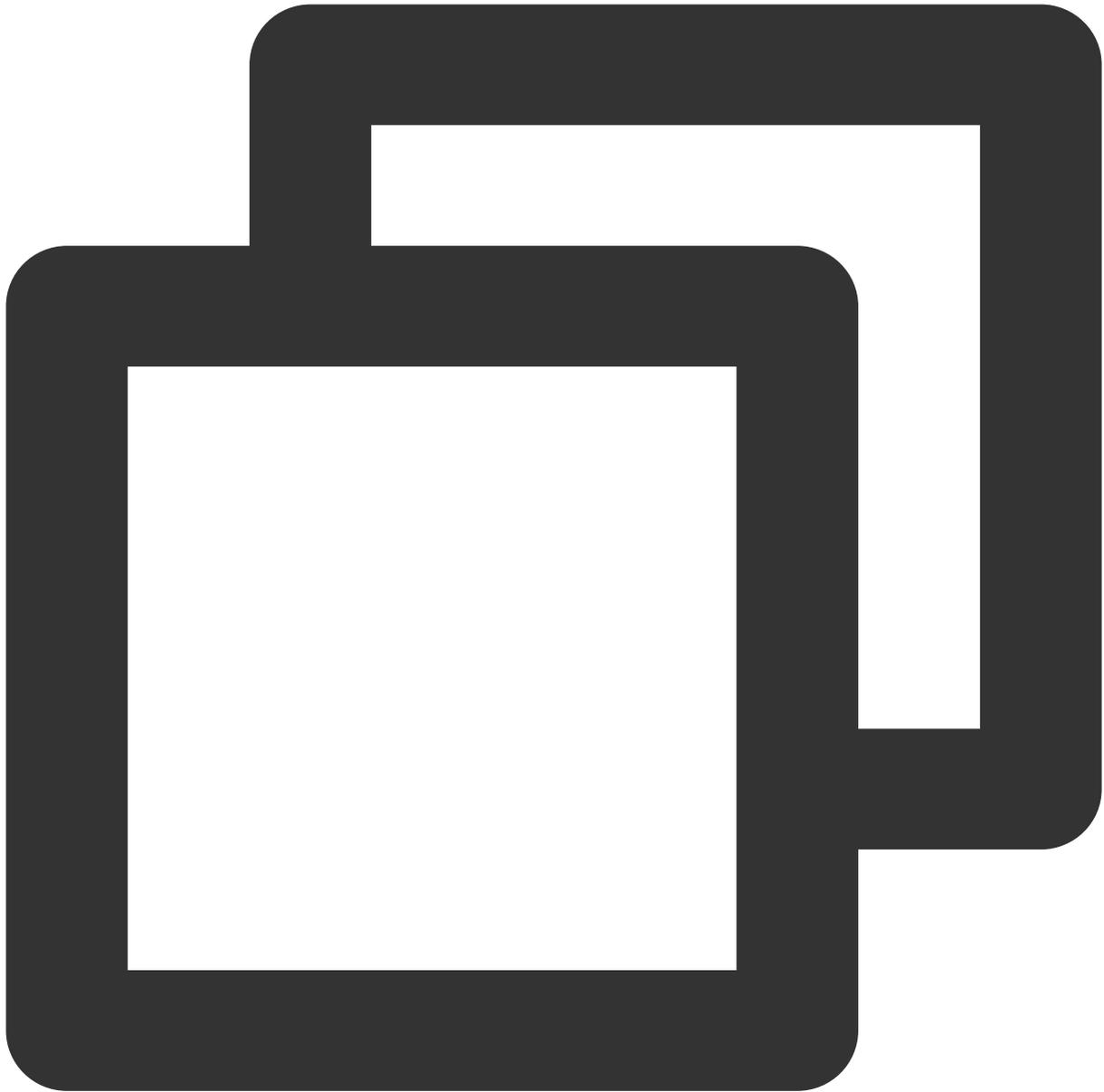
```
package main

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

```
func main() {
    // 将 examplebucket-1250000000 和 COS_REGION 修改为真实的信息
    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",
        },
    })
    // 对象键 (Key) 是对象在存储桶中的唯一标识。
    // 例如, 在对象的访问域名 `examplebucket-1250000000.cos.COS_REGION.myqcloud.com/test
    name := "test.go"
    // 1.通过字符串上传对象
    f := strings.NewReader("Hello COS")

    response, err := c.Object.Put(context.Background(), name, f, nil)
    if err != nil {
        // error 信息中直接包含 RequestId 字段
        panic(err)
    }
    requestId := response.Header.Get("X-Cos-Request-Id")
    fmt.Println(requestId)
}
```

通过 Java SDK 获取

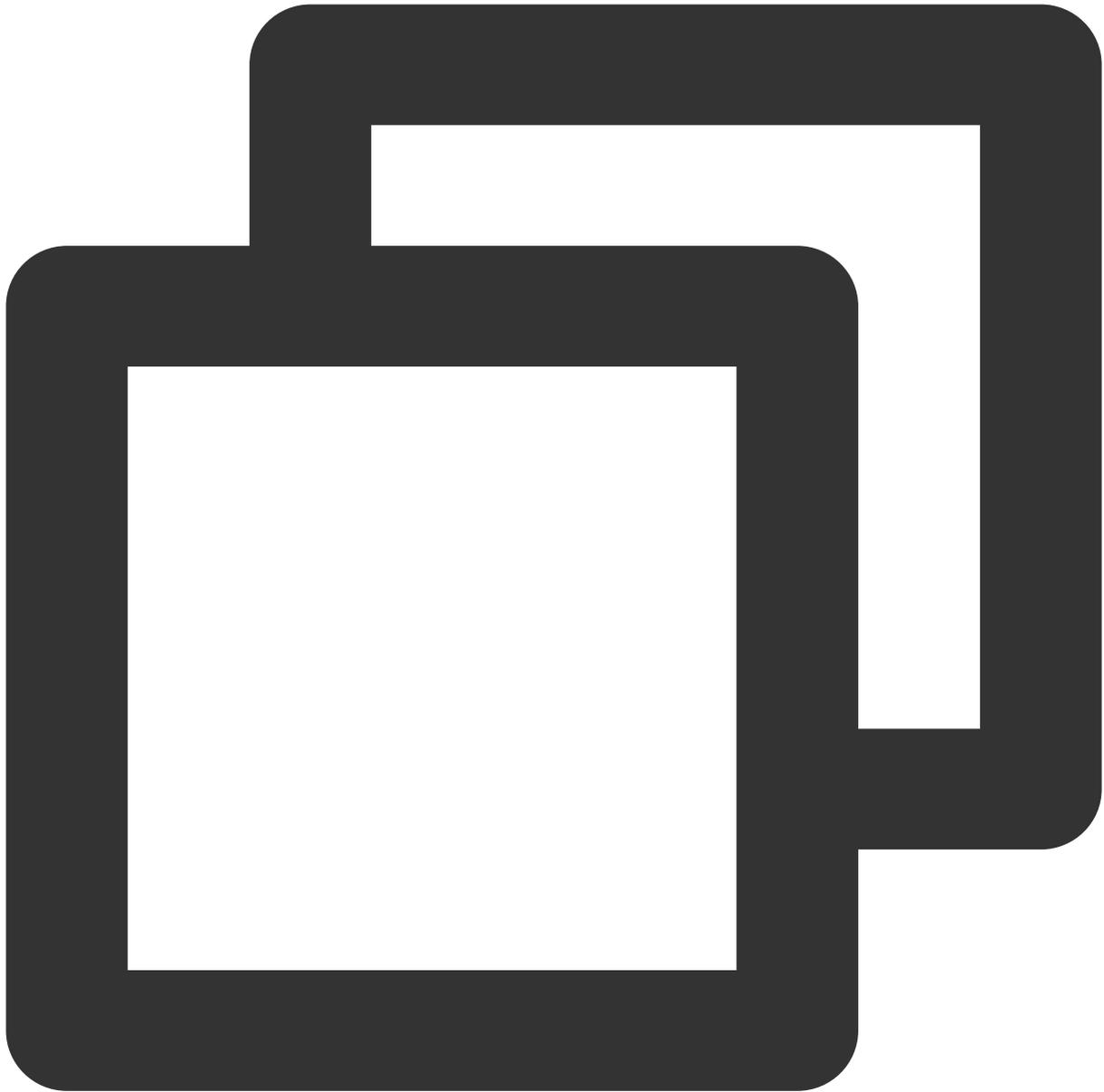


```
// 1 初始化用户身份信息 (secretId, secretKey) 。
String secretId = "SECRETID";
String secretKey = "SECRETKEY";
COSCredentials cred = new BasicCOSCredentials(secretId, secretKey);
// 2 设置 bucket 的地域, COS 地域的简称请参照 https://cloud.tencent.com/document/product
// clientConfig 中包含了设置 region, https(默认 http), 超时, 代理等 set 方法, 使用可参见源
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// 这里建议设置使用 https 协议
clientConfig.setHttpProtocol(HttpProtocol.https);
// 3 生成 cos 客户端。
```

```
COSClient cosClient = new COSClient(cred, clientConfig);
// Bucket 的命名格式为 BucketName-APPID , 此处填写的存储桶名称必须为此格式
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);
```

通过 Python SDK 获取



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

# 正常情况日志级别使用 INFO，需要定位时可以修改为 DEBUG，此时 SDK 会打印和服务端的通信信息
logging.basicConfig(level=logging.INFO, stream=sys.stdout)

# 1. 设置用户属性，包括 secret_id, secret_key, region等。Appid 已在 CosConfig 中移除，请在
```

```
secret_id = os.environ['COS_SECRET_ID']      # 用户的 SecretId, 建议使用子账号密钥, 授权遵
secret_key = os.environ['COS_SECRET_KEY']    # 用户的 SecretKey, 建议使用子账号密钥, 授权遵
region = 'ap-beijing'                       # 替换为用户的 region, 已创建桶归属的 region 可以在控制台查看, r
                                             # COS 支持的所有 region 列表参见 https://cloud.tencent.com/
token = None                                 # 如果使用永久密钥不需要填入 token, 如果使用临时密钥需要填入, 临时
scheme = 'https'                             # 指定使用 http/https 协议来访问 COS, 默认为 https, 可不填

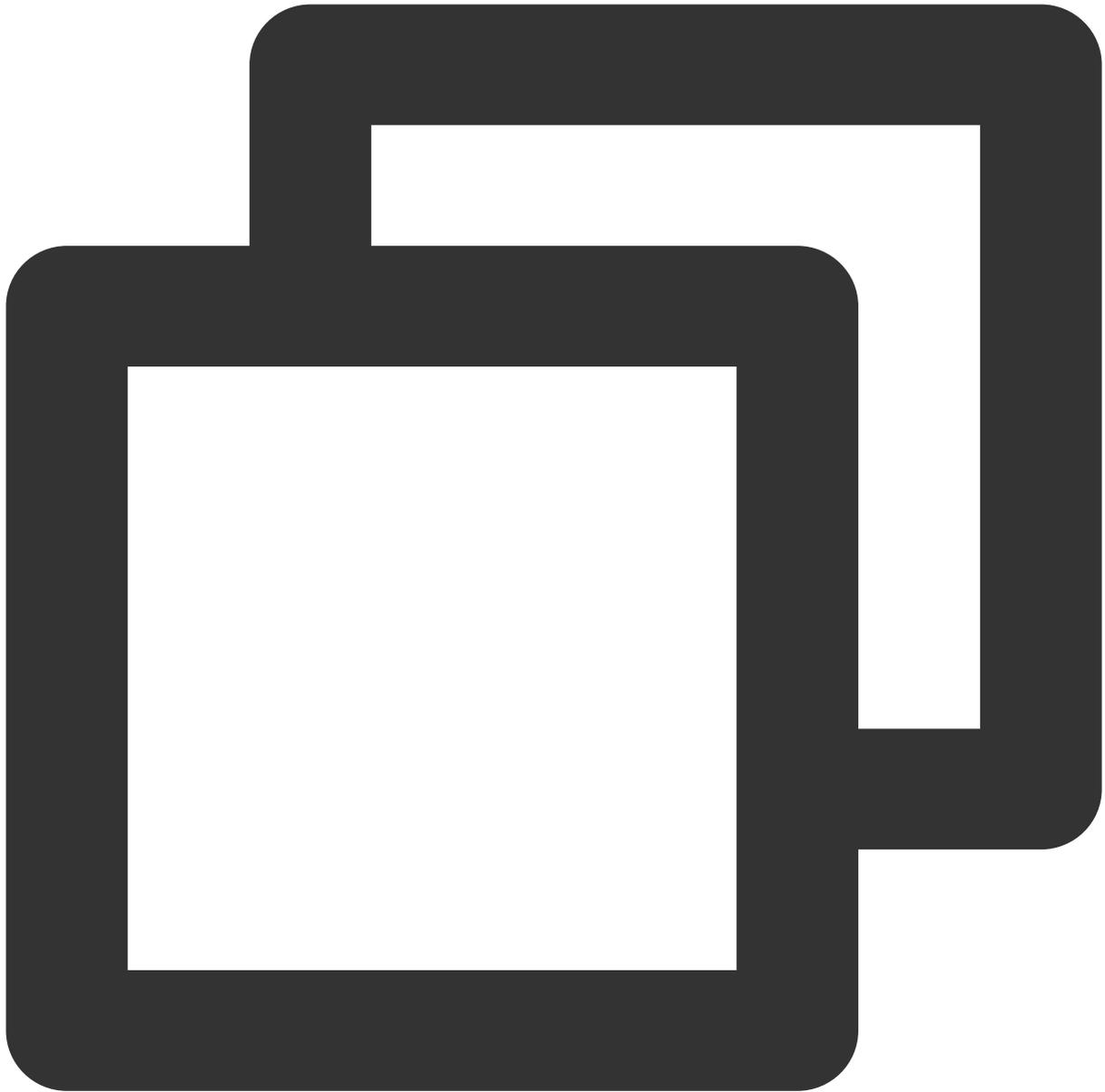
config = CosConfig(Region=region, SecretId=secret_id, SecretKey=secret_key, Token=t
client = CosS3Client(config)

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

    # 请求正常返回通过 response 查看 request-id
    if 'x-cos-request-id' in response:
        print(response['x-cos-request-id'])

# 请求失败通过异常查看 request-id
except CosServiceError as e:
    print(e.get_request_id())
```

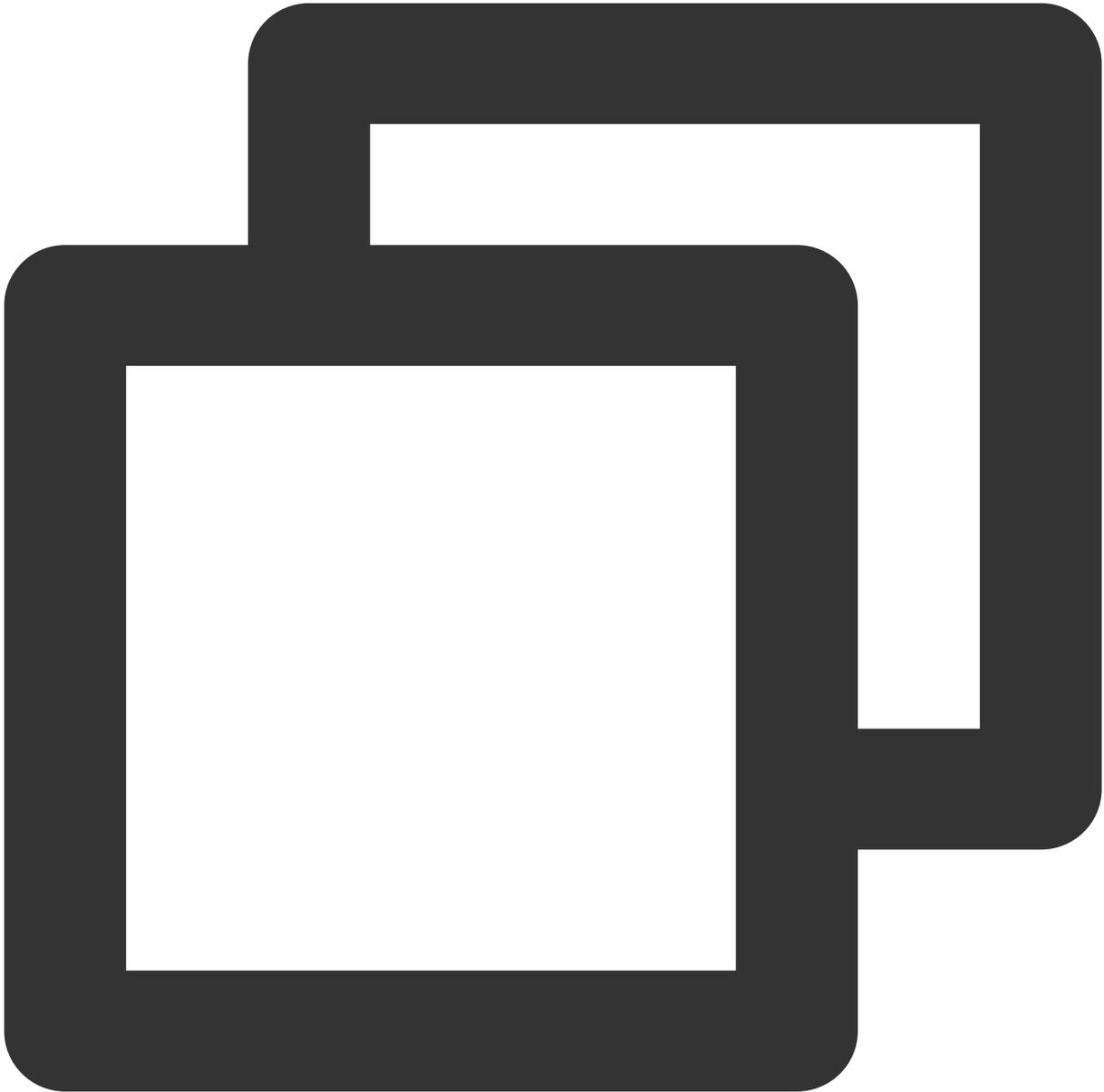
通过 JavaScript SDK 获取



```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 必须 */
  Region: 'COS_REGION',      /* 必须 */
  Key: 'test.js',            /* 必须 */
  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 );  
});
```

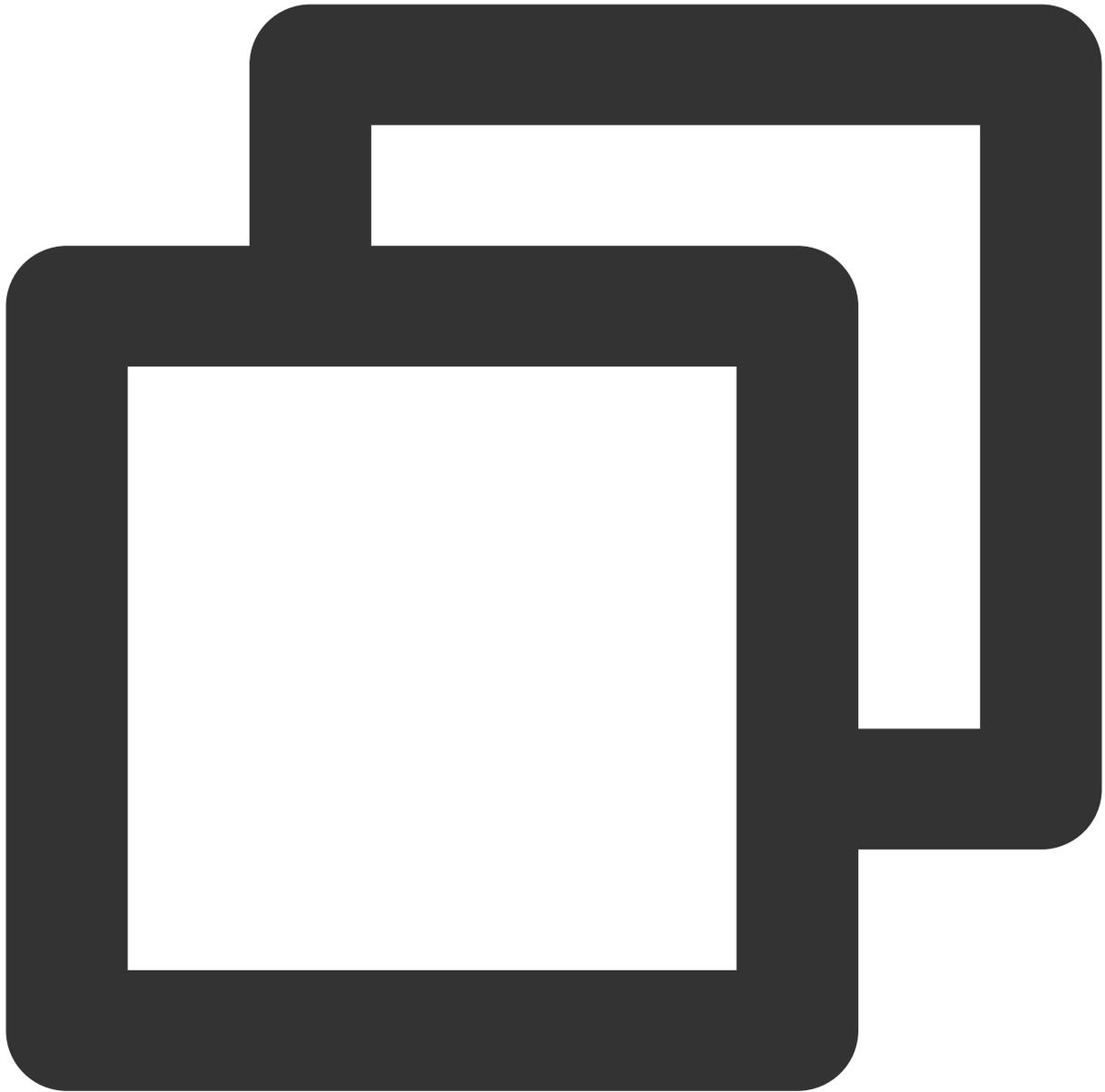
通过 Node.js SDK 获取



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

```
cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 必须 */
  Region: 'COS_REGION', /* 必须 */
  Key: 'test.nodejs', /* 必须 */
  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 );
});
```

通过 微信小程序 SDK 获取

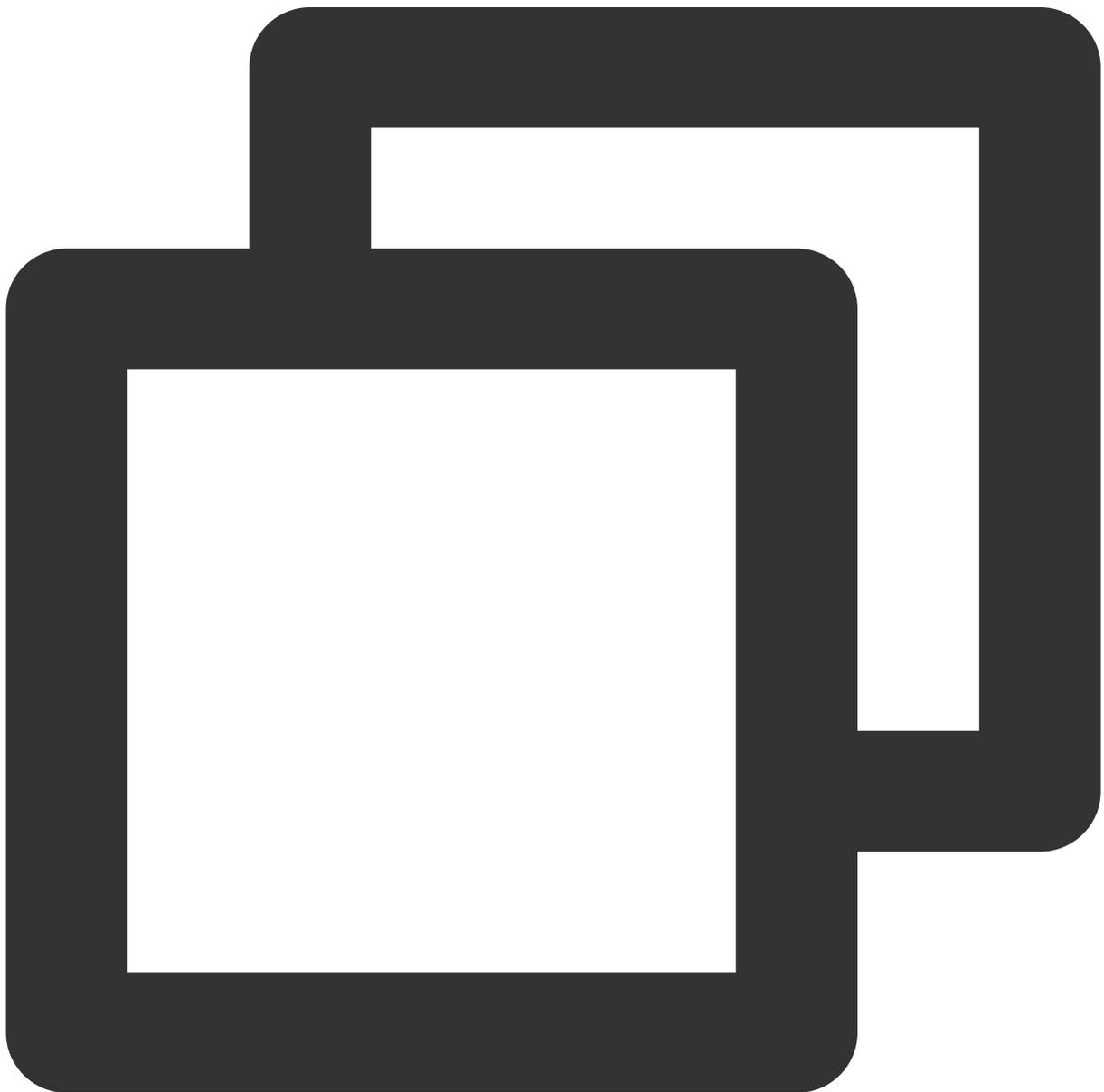


```
var COS = require('cos-wx-sdk-v5');
var cos = new COS({
  SecretId: 'SECRETID',
  SecretKey: 'SECRETKEY'
});

cos.putObject({
  Bucket: 'examplebucket-1250000000', /* 必须 */
  Region: 'COS_REGION', /* 必须 */
  Key: 'test.js', /* 必须 */
  StorageClass: 'STANDARD',
```

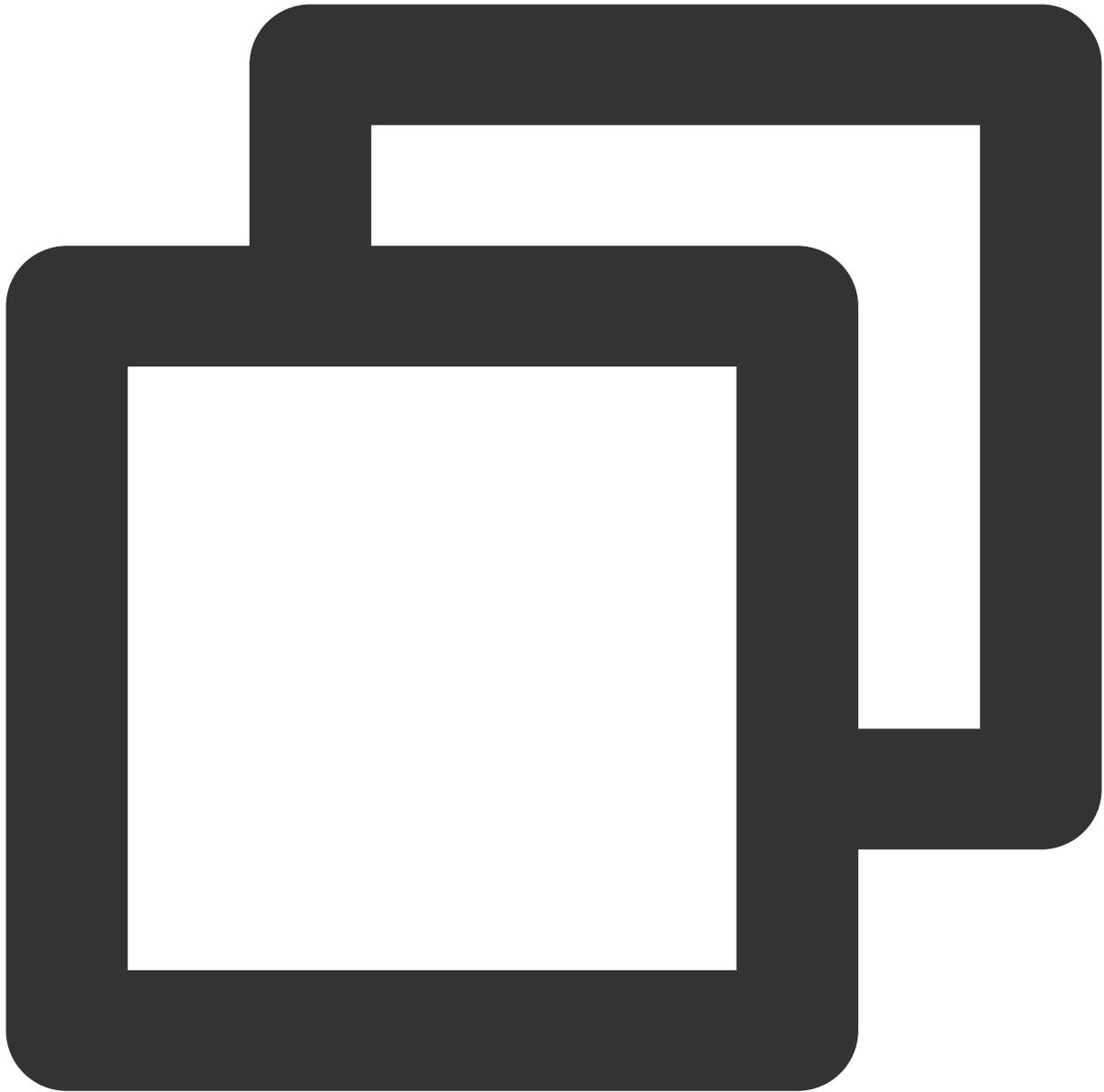
```
Body: 'Hello COS',
onProgress: function(progressData) {
    console.log(JSON.stringify(progressData));
}
}, function(err, data) {
    var requestId = (err || data).headers['x-cos-request-id'];
    console.log(requestId );
});
```

通过 PHP SDK 获取



```
$secretId = "SECRETID"; //云 API 密钥 SecretId";
$secretKey = "SECRETKEY"; //云 API 密钥 SecretKey";
$region = "COS_REGION"; //设置一个默认的存储桶地域
$cosClient = new Qcloud\Cos\Client(
    array(
        'region' => $region,
        'schema' => 'https', //协议头部, 默认为 http
        'credentials'=> array(
            'secretId' => $secretId ,
            'secretKey' => $secretKey));
# 上传文件
## putObject (上传接口, 最大支持上传5G文件)
### 上传内存中的字符串
try {
    $bucket = "examplebucket-1250000000"; //存储桶名称 格式: BucketName-APPID
    $key = "test.php"; //此处的 key 为对象键, 对象键是对象在存储桶中的唯一标识
    $result = $cosClient->putObject(array(
        'Bucket' => $bucket,
        'Key' => $key,
        'Body' => 'Hello COS'));
    $requestId = $result['RequestId'];
    print_r($requestId);
} catch (\Exception $e) {
    echo "$e\n";
}
```

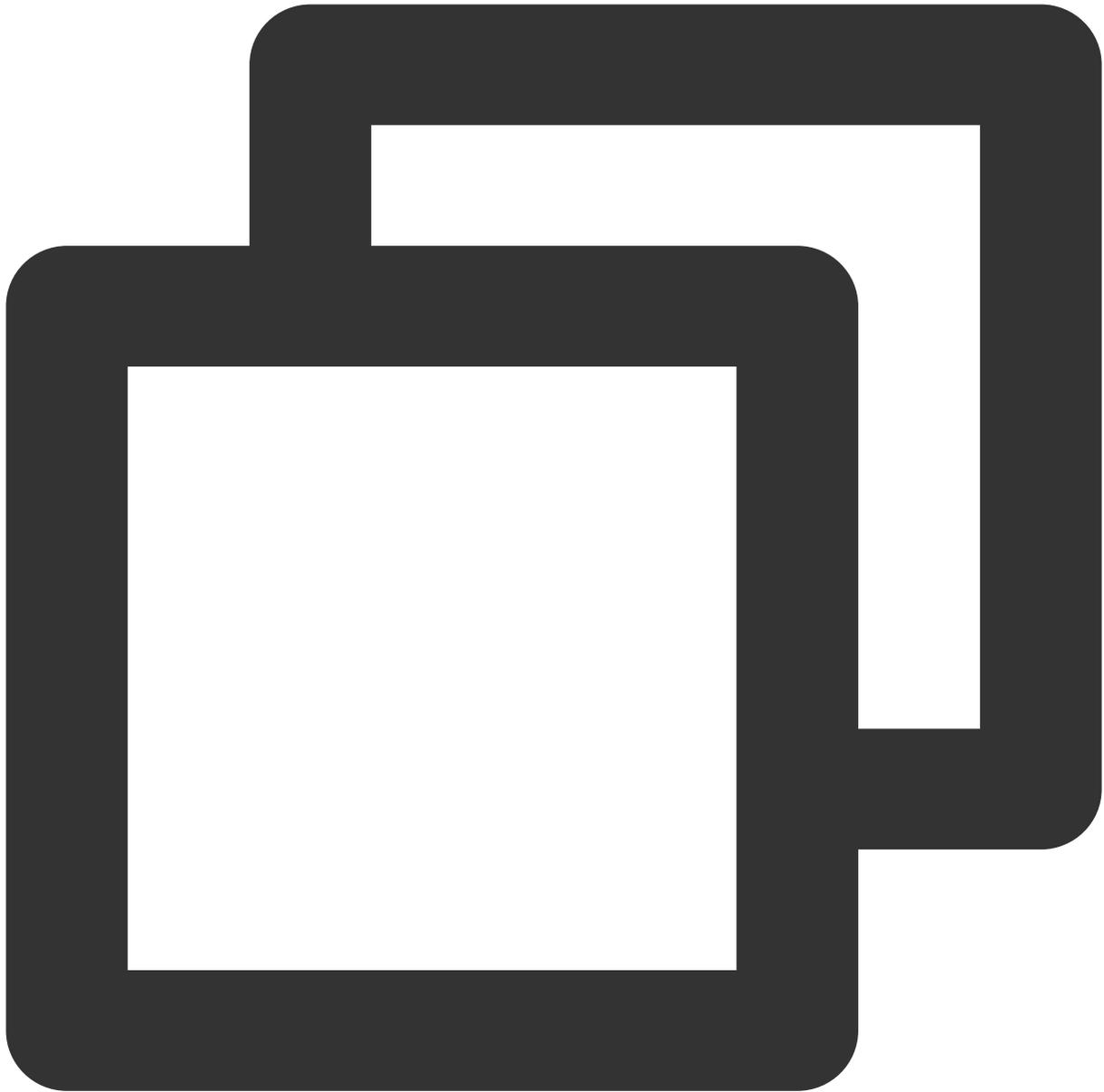
通过 iOS SDK 获取



```
QCloudCOSXMLUploadObjectRequest* put = [QCloudCOSXMLUploadObjectRequest new];
/** 本地文件路径, 请确保 URL 是以 file:// 开头, 格式如下 :
1. [NSURL URLWithString:@"file:///var/mobile/Containers/Data/Application/DBPF7490-
2. [NSURL fileURLWithPath:@"/var/mobile/Containers/Data/Application/DBPF7490-D5U8-4
*/
NSURL* url = [NSURL fileURLWithPath:@"文件的URL"];
// 存储桶名称, 由 BucketName-Appid 组成, 可以在 COS 控制台查看 https://console.tencentclo
put.bucket = @"examplebucket-1250000000";
// 对象键, 是对象在 COS 上的完整路径, 如果带目录的话, 格式为 "video/xxx/movie.mp4"
put.object = @"exampleobject";
// 需要上传的对象内容. 可以传入 NSData*或者 NSURL*类型的变量
```

```
put.body = url;
// 监听上传进度
[put setSendProcessBlock:^(int64_t bytesSent,
                          int64_t totalBytesSent,
                          int64_t totalBytesExpectedToSend) {
    //      bytesSent          本次要发送的字节数（一个大文件可能要分多次发送）
    //      totalBytesSent     已发送的字节数
    //      totalBytesExpectedToSend 本次上传要发送的总字节数（即一个文件大小）
}];
// 监听上传结果
[put setFinishBlock:^(QCloudUploadObjectResult *result, NSError *error) {
    // 获取requestid
    [result.__originHTTPURLResponse__.allHeaderFields objectForKey:@"x-cos-request-id"];
}];
[put setInitMultipleUploadFinishBlock:^(QCloudInitiateMultipartUploadResult *
                                       multipleUploadInitResult,
                                       QCloudCOSXMLUploadObjectResumeData resumeData) {
    // 在初始化分块上传完成以后会回调该 block, 在这里可以获取 resumeData, uploadid
    NSString* uploadId = multipleUploadInitResult.uploadId;
}];
[[QCloudCOSTransferMangerService defaultManager] UploadObject:put];
```

通过 Android SDK 获取



```
// 1. 初始化 TransferService。在相同配置的情况下，您应该复用同一个 TransferService
TransferConfig transferConfig = new TransferConfig.Builder()
    .build();
CosXmlServiceConfig cosXmlServiceConfig = new CosXmlServiceConfig.Builder()
    .setRegion(COS_REGION)
    .builder();
CosXmlService cosXmlService = new CosXmlService(context, cosXmlServiceConfig, creden
TransferService transferService = new TransferService(cosXmlService, transferConfig

// 2. 初始化 PutObjectRequest
String bucket = "examplebucket-1250000000"; //存储桶，格式：BucketName-APPID
```

```
String cosPath = "exampleobject"; //对象在存储桶中的位置标识符，即称对象键
String srcPath = "examplefilepath"; //本地文件的绝对路径
PutObjectRequest putObjectRequest = new PutObjectRequest(bucket,
    cosPath, srcPath);

// 3. 调用 upload 方法上传文件
final COSUploadTask uploadTask = transferService.upload(putObjectRequest);
uploadTask.setCosXmlResultListener(new CosXmlResultListener() {
    @Override
    public void onSuccess(CosXmlRequest request, CosXmlResult result) {
        // 上传成功，可以在这里拿到 requestId
        String requestId = result.getHeader("x-cos-request-id");
    }

    @Override
    public void onFail(CosXmlRequest request,
        CosXmlClientException clientException,
        CosXmlServiceException serviceException) {
        // 只有 CosXmlServiceException 异常才会有 requestId
        if (serviceException != null) {
            String requestId = serviceException.getRequestId();
        }
    }
});
```

通过外网上传文件至 COS 缓慢

最近更新时间：2024-01-06 11:27:57

现象描述

现象1：

使用公司网络进行上传时，传输正常；使用家庭网络进行上传时，传输缓慢（低于8Mbps）。
使用手机4G网络进行上传时，传输正常；使用公司网络进行上传时，传输缓慢（低于8Mbps）。

现象2：使用自定义域名进行上传时，传输缓慢。

可能原因

针对现象1：

1.1 如果您在不同的网络环境下访问 COS 的速率不同，可能和当前网络运营商及网络环境有关。

1.2 如果您在不同的网络环境下访问 COS 的速率不同，可能因跨境访问导致。

针对现象2：自定义域名 CNAME 到其他产品再转回到 COS，如内容分发网络（Content Delivery Network, CDN）、云服务器（Cloud Virtual Machine, CVM）、安全高防产品。

解决思路

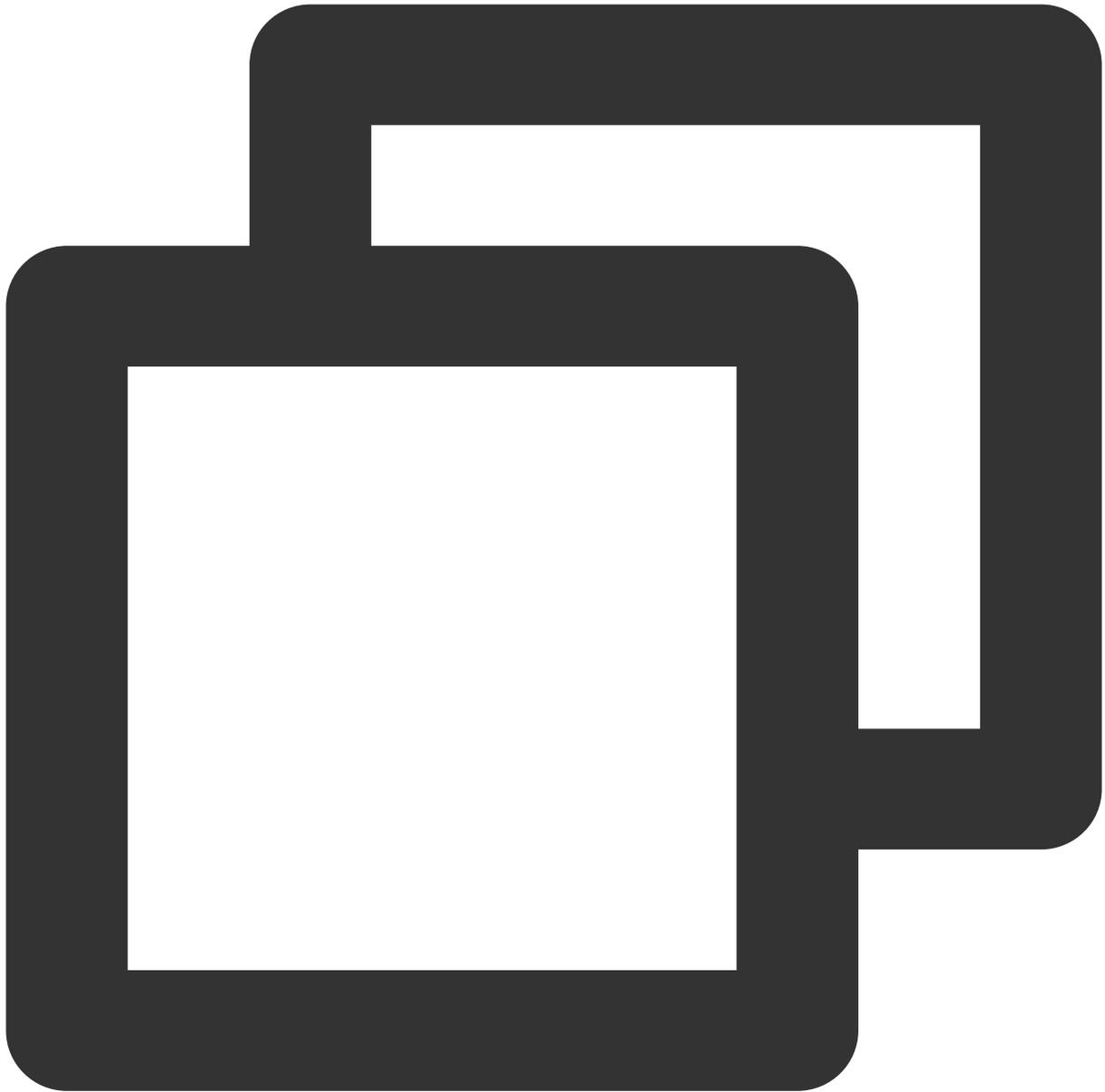
如果您遇到 [现象1](#) 的情况，则可以通过检查客户端网络环境的方式自行处理。操作详情请参见 [排查客户端网络](#)。

如果您遇到 [现象2](#) 的情况，则可以通过修改自定义域名解析的方式减少传输中转链路，提高传输效率。操作详情请参见 [修改自定义域名解析](#)。

处理步骤

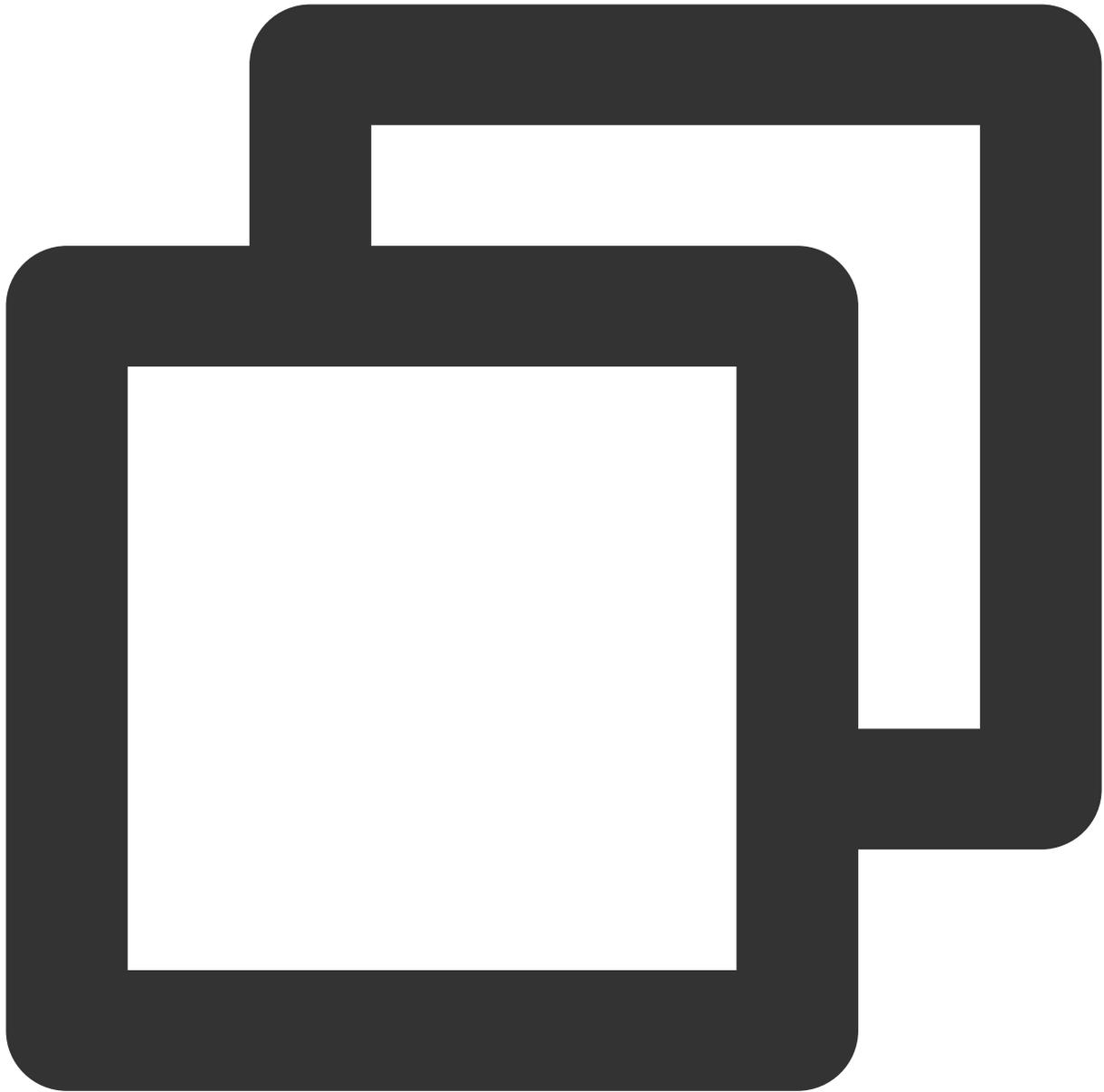
排查客户端网络

1. 执行以下命令，确认 IP 地址运营商与客户端网络运营商是否相符。



ping COS 的访问域名

例如：



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

是，请执行 [步骤3](#)。

否，请执行 [步骤2](#)。

2. 以 Chrome 浏览器为例，检查浏览器是否设置了代理。

2.1 打开 Chrome 浏览器，单击右上角的

>  **设置**，打开设置页面。

2.2 单击**高级**，在“系统”栏中选择**在您计算机的代理设置**，打开操作系统的设置窗口。

检查是否设置了代理。

是，关闭代理。

否，请执行 [步骤3](#)。

3. 检查所用的 Wi-Fi 路由器是否存在限速。

是，请根据实际需求，酌情放行。

否，请执行 [步骤4](#)。

4. 检查当前网络上传 COS 的传输性能。

以 COS 的 COSCMD 工具为例，测试一个20MB对象的上传和下载性能。



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

返回类似如下结果，分别得出平均速率（Average），最低速率（Min），最高速率（Max）。

```
D:\work>coscmd probe -n 1 -s 20
Upload tmp_test_20M => cos://[redacted]/tmp_
[success]
Download cos://[redacted]/tmp_test_20M => tm
[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
```

5. 通过浏览器访问 [测速网](#)，并结合 [步骤4](#) 检查客户端的网络带宽占用率是否达到上限。

如果步骤4的速率低于客户端带宽速率，请 [联系我们](#)。

如果步骤4的速率等于客户端带宽速率，且未达到运营商承诺的带宽，请联系运营商客服。

如果步骤4的速率等于客户端带宽速率，且达到了运营商承诺的带宽，请执行 [步骤6](#)。

6. 检查是否存在国内客户端访问海外节点 bucket，或者存在海外客户端访问国内节点 bucket。

是，建议使用 COS 的全球加速功能。

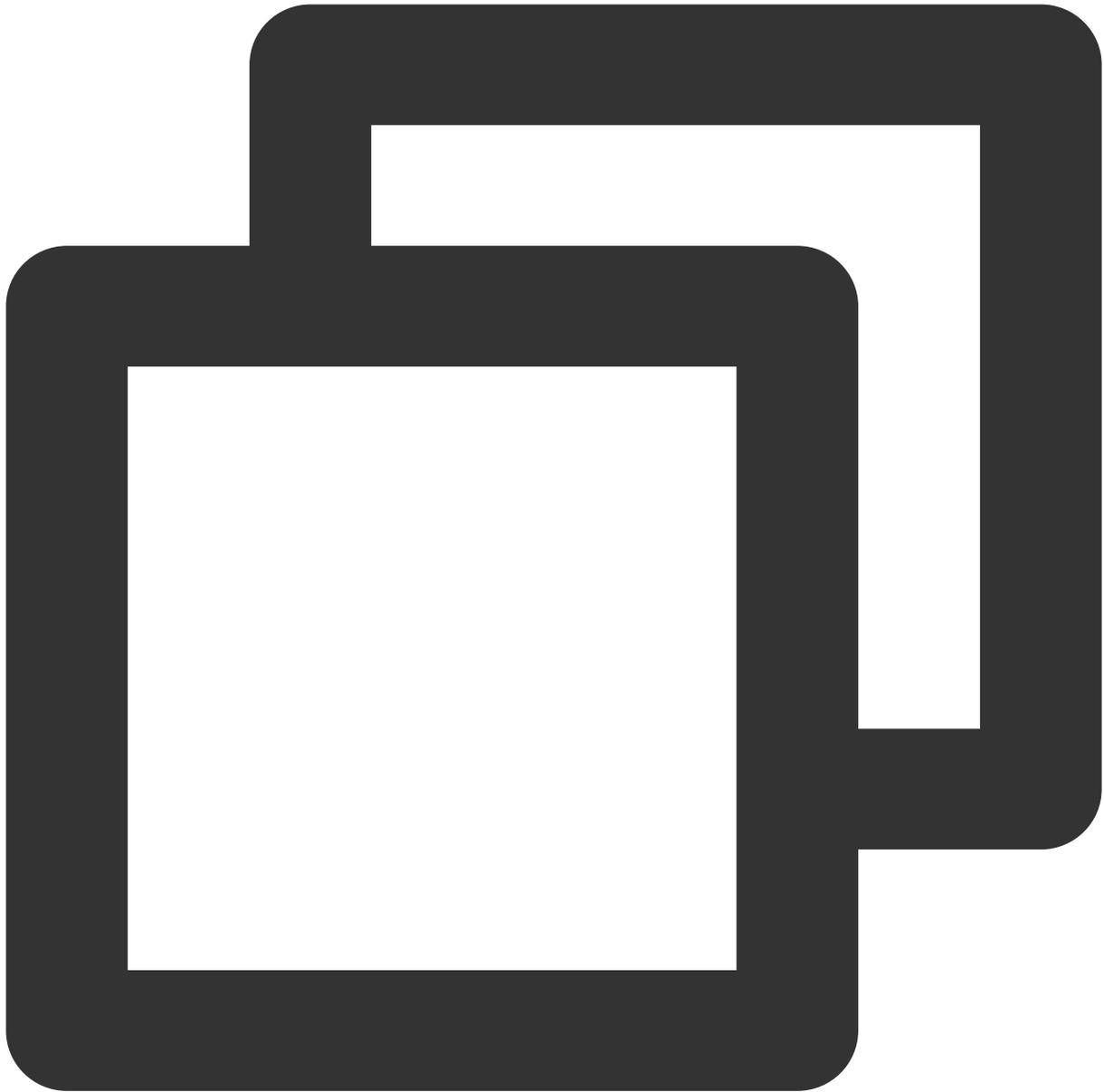
否，请 [联系我们](#)。

修改自定义域名解析

1. 检查自定义域名解析是否为 COS 域名。

是，请 [联系我们](#)。

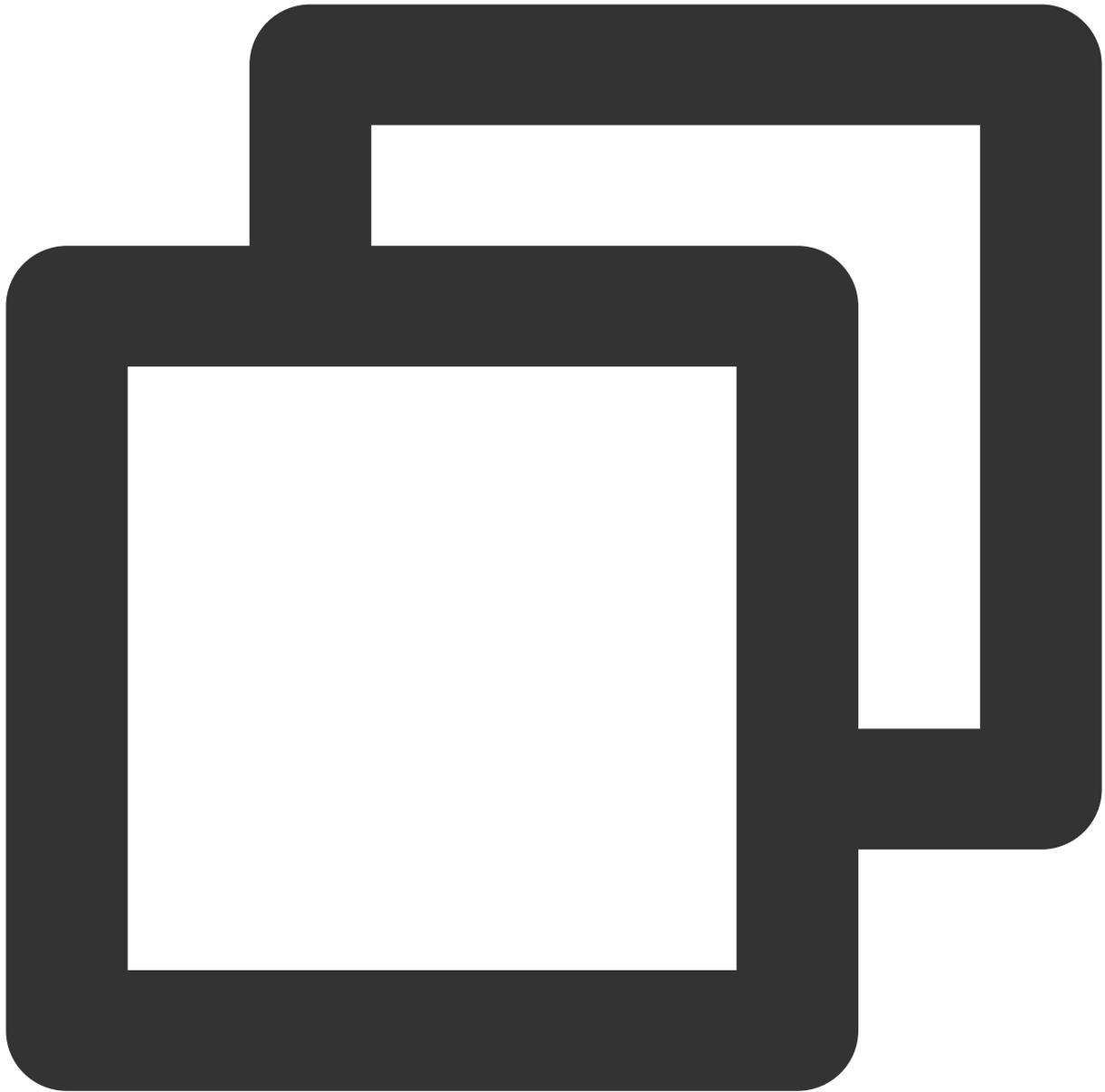
常见的 COS 域名如下：



```
XXX.cos.ap-beijing.myqcloud.com (COS 默认域名)  
XXX.cos.accelerate.myqcloud.com (COS 全球加速域名)  
XXX.cos-website.ap-beijing.myqcloud.com (COS 静态页域名)  
XXX.picbj.myqcloud.com (COS 数据万象默认域名)
```

否，请执行 [步骤2](#)。

常见的非 COS 域名如下：



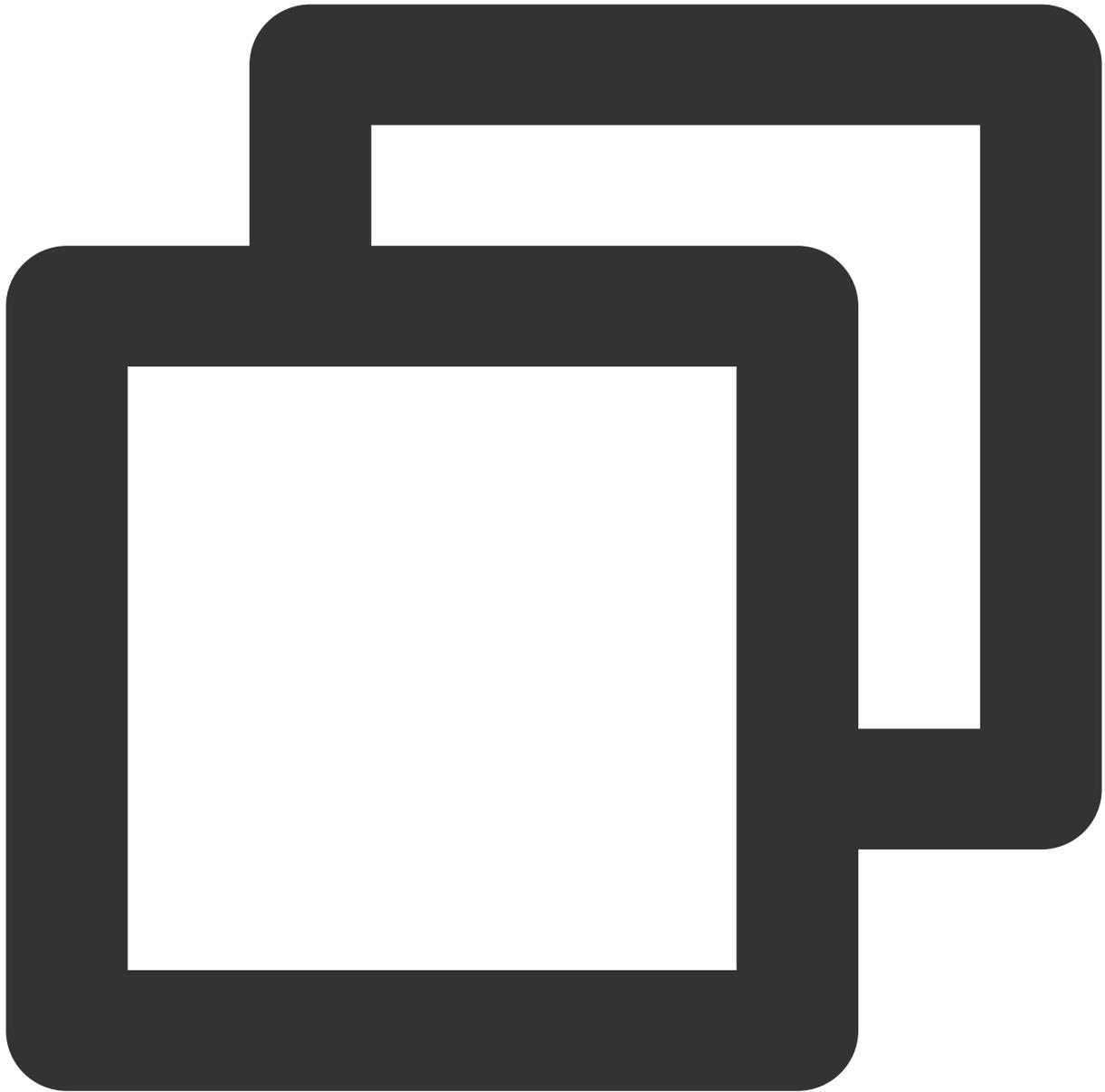
XXX.file.myqcloud.com 或 XXX.cdn.dnsv1.com (腾讯云 CDN 默认域名)
XXX.w.kunlungr.com (aliyunCDN 默认域名)

2. 将自定义域名的 CNAME 解析到所需的 COS 域名中，并进行数据上传。

例如 `upload.mydomain.com cname XXX.cos.ap-beijing.myqcloud.com`，具体操作请参见 [开启自定义源站域名](#)。

3. 修改客户端的默认 COS 域名。

以 C# 代码为例：



```
CosXmlConfig config = new CosXmlConfig.Builder()
    .SetConnectionTimeoutMs(60000) //设置连接超时时间, 单位 毫秒 , 默认 45000ms
    .SetReadWriteTimeoutMs(40000) //设置读写超时时间, 单位 毫秒 , 默认 45000ms
    .IsHttps(true) //设置默认 https 请求
    .SetAppid(appid) //设置腾讯云账户的账户标识 APPID
    .SetRegion(region) //设置一个默认的存储桶地域
    .SetHost("XXXXXX.com") //输入自定义域名
    .SetDebugLog(true) .Build(); //创建 CosXmlConfig 对象
```

其他 SDK 调用请参见 [SDK 概览](#)。

访问 COS 时返回403错误码

最近更新时间：2024-01-06 11:27:57

故障现象

当 COS API、SDK 上传和下载资源时，返回403错误码。

当使用临时密钥或子账号访问 COS 资源时，返回403错误码。

当修改 COS bucket 配置时，返回403错误码。

故障分析思路

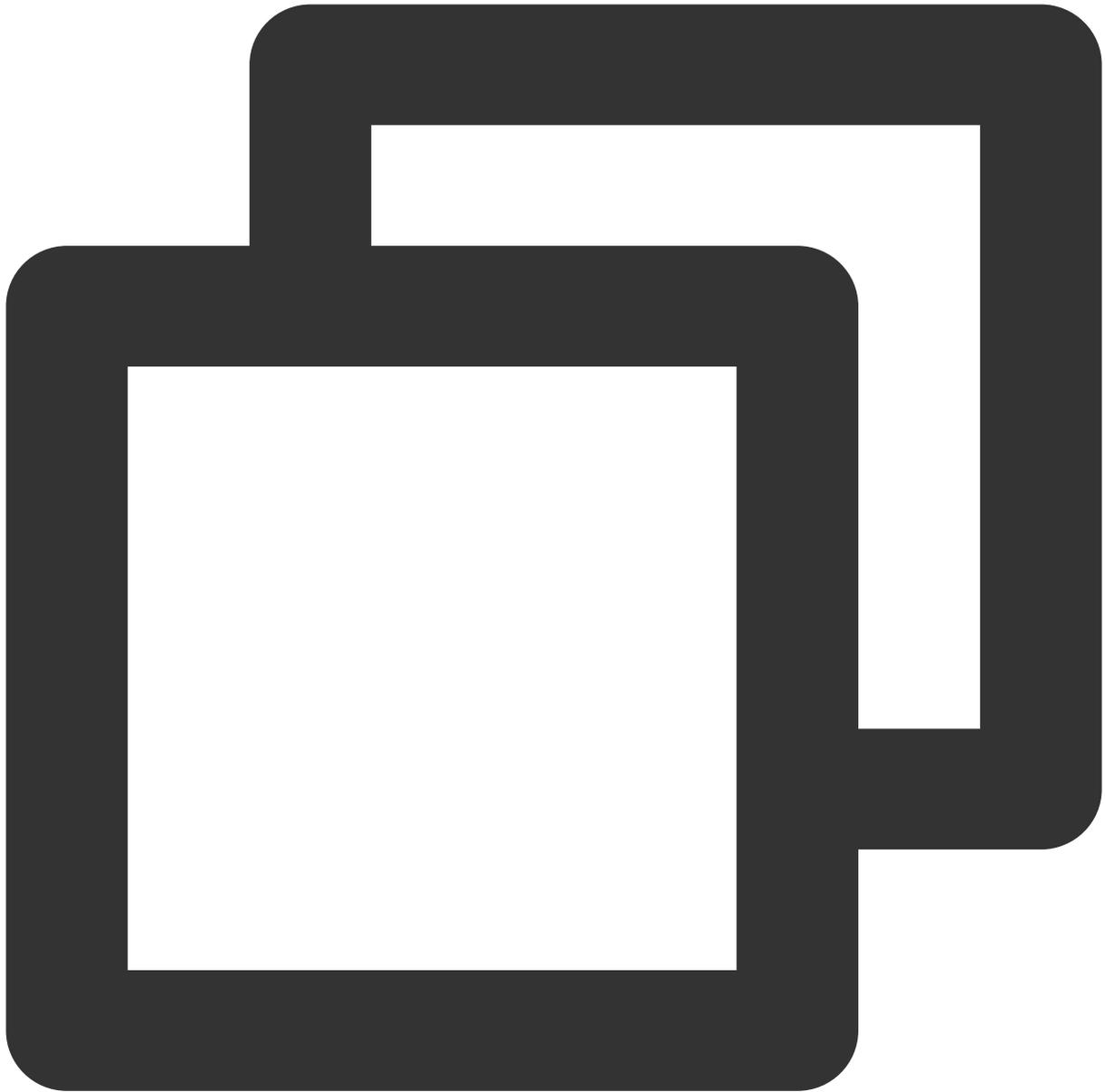
当 COS 请求出现403错误码时，可参考以下流程排查问题原因：

1. 检查请求是否为 CORS 跨域访问请求，失败的跨域请求会返回 "[AccessForbidden](#)"。
2. 检查请求是否命中存储桶防盗链配置，因防盗链规则引起的 403 错误码，COS 会返回 "[You are denied by bucket referer rule](#)"。
3. 检查请求是否是匿名请求，向非公有读的对象发起不带签名的请求，会返回 "Access Denied."，如要为存储桶或对象设置公有读，参考 [设置访问权限](#) 或 [设置对象的访问权限](#)。
4. 检查请求密钥和请求签名是否正确。
 - 4.1 当签名未正确使用 SecretId 时，COS 会返回 "[InvalidAccessKeyId](#)"。
 - 4.2 当本机时间不准，或请求时间超出签名有效期时，COS 会返回 "[RequestTimeTooSkewed](#)" 或 "[Request has expired](#)"。
 - 4.3 当生成签名的计算方式存在问题时，COS 会返回 "[SignatureDoesNotMatch](#)"。
5. 检查发起请求的子账号或临时密钥是否授予了相应访问权限。
 - 5.1 检查子账号的访问权限，未授予资源访问权限的子账号请求，会返回 "[Access Denied.](#)"。
 - 5.2 对于使用临时密钥发起的请求，申请临时密钥时填写的策略会限制临时密钥请求的资源范围，详见 [临时密钥生成与使用指引](#)。
6. 检查所请求对象是否为归档类型或深度归档类型，此类请求 COS 会返回 "[InvalidObjectState](#)"。

故障定位及处理

Message 为 "Access Denied."

当您访问 COS 出现如下信息时：



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

需要您执行以下操作：

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，选择**存储桶列表**，进入存储桶管理页面。
3. 找到需要操作的存储桶，单击该存储桶名称，进入存储桶配置页面。
4. 在左侧导航栏中，选择**权限管理 > 存储桶访问权限**，进入存储桶访问权限管理页面。
5. 在“存储桶访问权限”栏中，检查访问 COS 的账号是否配置了访问权限。

是，请执行下一步。

否，请单击[添加用户](#)，为访问 COS 的账号设置所需权限。

6. 检查配置访问权限的账号是否具有所需权限。

是，请执行下一步。

否，请单击[编辑](#)，重新设置。

7. 在“Policy权限设置”栏中，检查访问 COS 的账号是否配置了 policy 授权策略。

注意

如果存储桶访问权限为私有读写，且 Policy 权限为匿名访问，那么 Policy 权限的优先级高于存储桶访问权限。

在 Policy 授权策略中，如果同一个子用户同时设置了允许和禁止策略，那么禁止策略的优先级高于允许策略。

在 Policy 授权策略中，“所有用户”策略的优先级低于“指定用户”策略。

是，请执行下一步。

否，请单击[添加策略](#)，根据实际签名访问时账号所需权限进行设置。

8. 检查配置 Policy 权限的账号是否具有所需权限。

是，请执行下一步。

否，请单击[编辑](#)，重新设置。

9. 检查访问 COS 资源时使用的 q-ak 参数是否为目标桶的所属账号（区分大小写）。

是，请执行下一步。

否，请将 q-ak 参数修改为相应的目标桶所属账号。

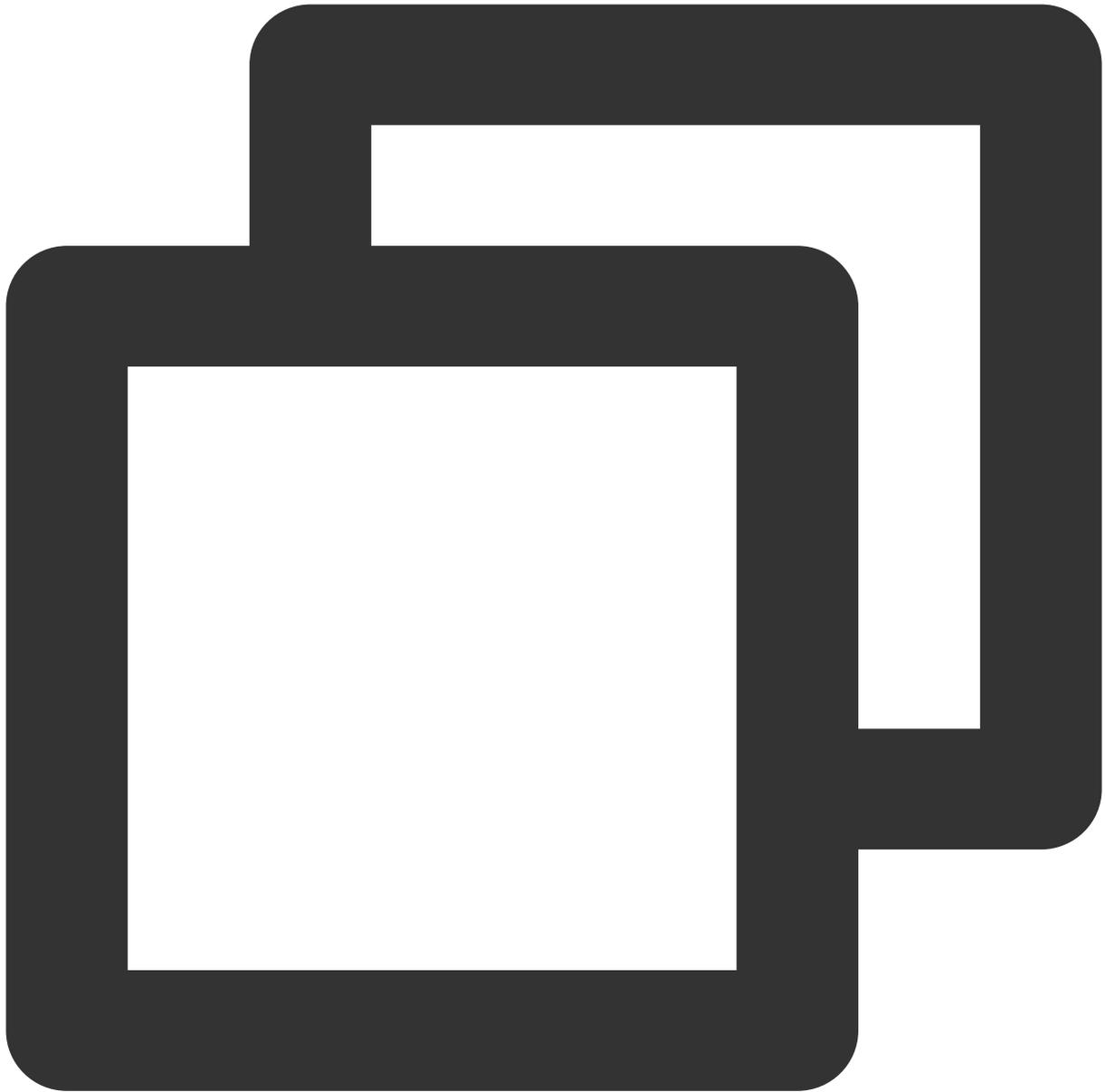
10. 检查访问 COS 资源时是否为跨账号访问。

是，请为该账号进行跨账号授权，详细操作请参见[授权跨账号的子账号对指定文件的读写权限](#)。

否，请 [联系我们](#)。

Message 为“AccessForbidden”

当您访问 COS 出现如下信息时：



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

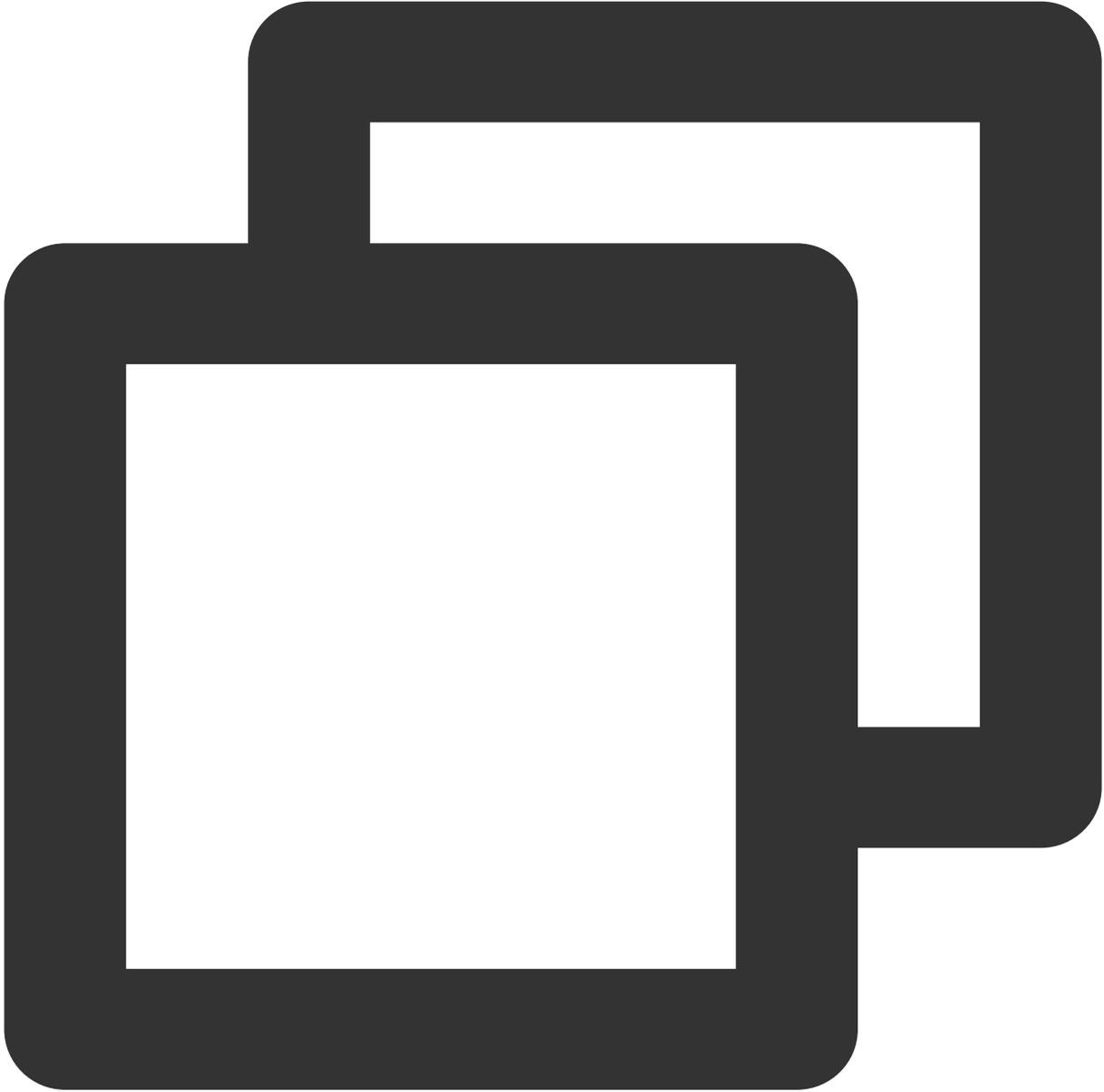
需要您执行以下操作：

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，选择**存储桶列表**，进入存储桶管理页面。
3. 找到需要操作的存储桶，单击该存储桶名称，进入存储桶配置页面。
4. 在左侧导航栏中，选择**安全管理 > 跨域访问CORS设置**，进入跨域访问 CORS 设置页面。
5. 在“跨域访问CORS设置”栏中，检查是否为跨域请求。

是，请执行下一步。

否，请修改规则。

6. 执行以下命令，检查跨域请求配置是否正确。



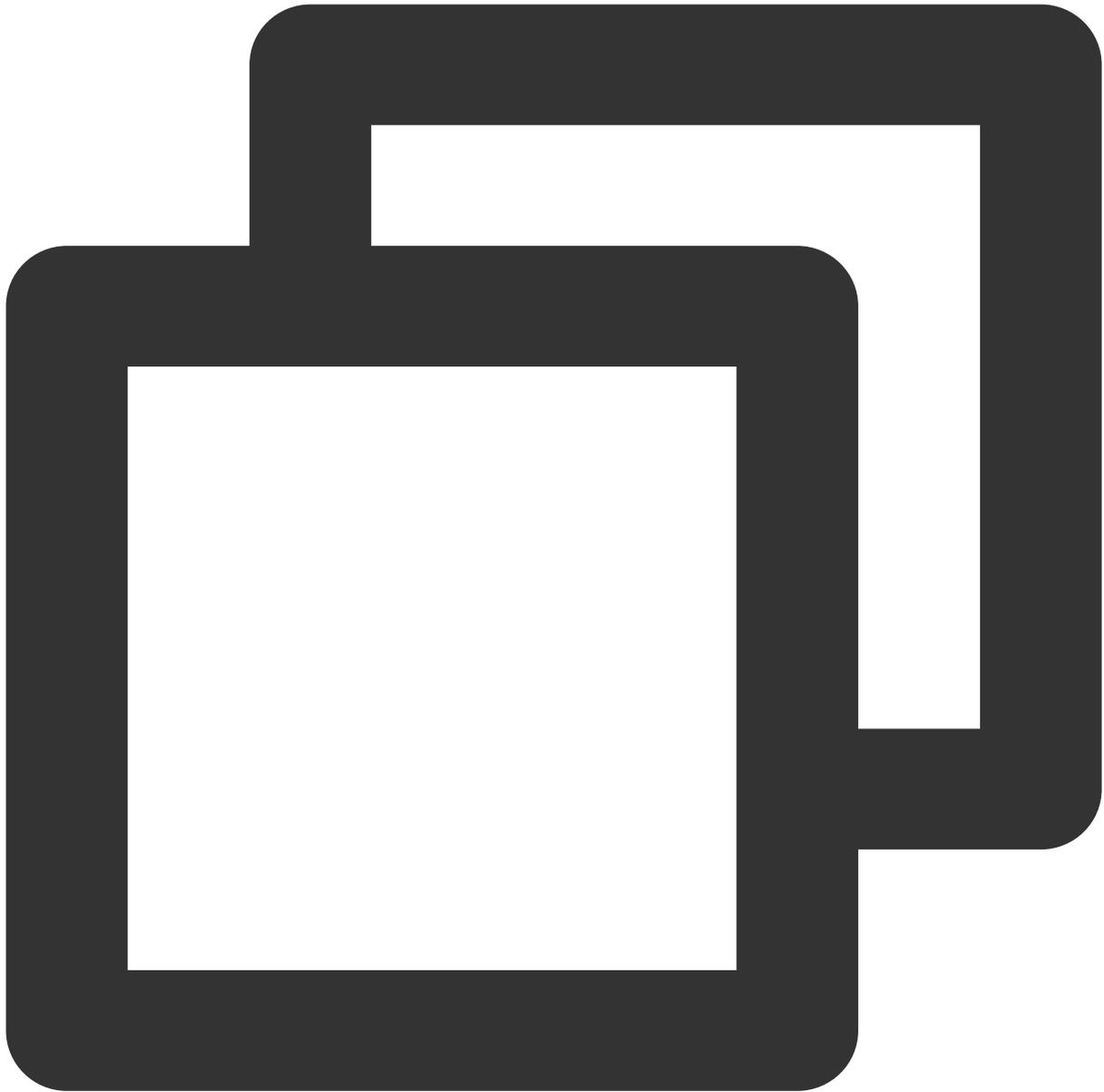
```
curl 'http://bucket-appid.cos.ap-guangzhou.myqcloud.com/object' -voa /dev/null -H '
```

返回如下信息，即表示配置正确。

```
(base) [root@VM_32_5_centos ~]# curl 'http://[redacted].cos.ap-guangzhou.myqcloud.com/1.gif' -voa /dev/null -H 'Origin: http://[redacted]'
% Total % Received % Xferd Average Speed Time Time Time Current
         Dload Upload Total Spent Left Speed
* Connected to [redacted].cos.ap-guangzhou.myqcloud.com ([redacted]) port 80 (#0)
> GET /1.gif HTTP/1.1
> Host: [redacted].cos.ap-guangzhou.myqcloud.com
> User-Agent: curl/7.29.0
> Accept: */*
> Origin: http://[redacted]
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: image/gif
< Content-Length: 307293
< Connection: keep-alive
< Accept-Ranges: bytes
< Access-Control-Allow-Credentials: true
< Access-Control-Allow-Methods: PUT,GET,POST
< Access-Control-Allow-Origin: http://[redacted]
< Access-Control-Expose-Headers: myheader
< Date: Tue, 30 Mar 2021 05:07:44 GMT
< ETag: "2c1f37c1cd1eb5da9a7df5774ed27a53"
< Last-Modified: Tue, 16 Mar 2021 14:03:51 GMT
< Server: tencent-cos
< x-cos-hash-crc64ecma: 63978585 [redacted]1
< x-cos-replication-status: Replica
< x-cos-request-id: NjA2Mjk2MDBfN2ExZ [redacted]4YTVm
< x-cos-version-id: MTg0NDUx [redacted]zA
<
{ [data not shown]
100 300k 100 300k 0 0 783k 0 --:--:-- --:--:-- --:--:-- 783k
* Connection #0 to host [redacted].cos.ap-guangzhou.myqcloud.com left intact
* Closing connection -1
curl: (3) URL using bad/illegal format or missing URL
```

Message 为 “You are denied by bucket referer rule”

当您访问 COS 出现如下信息时：



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

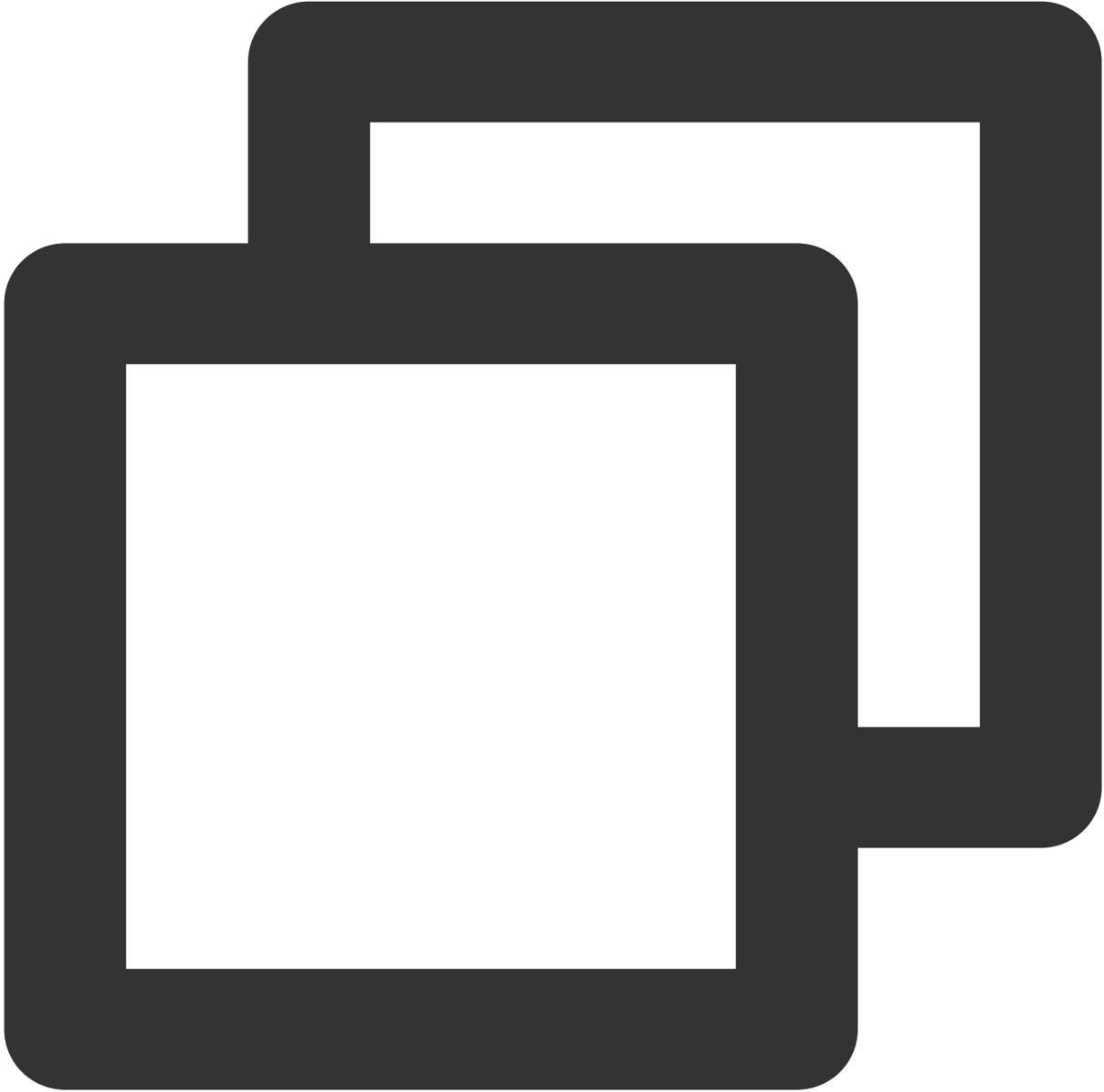
需要您执行以下操作：

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，选择**存储桶列表**，进入存储桶管理页面。
3. 找到需要操作的存储桶，单击该存储桶名称，进入存储桶配置页面。
4. 在左侧导航栏中，选择**安全管理 > 防盗链设置**，进入防盗链设置页面。
5. 在“防盗链设置”中，检查是否设置了防盗链。

是，请执行下一步。

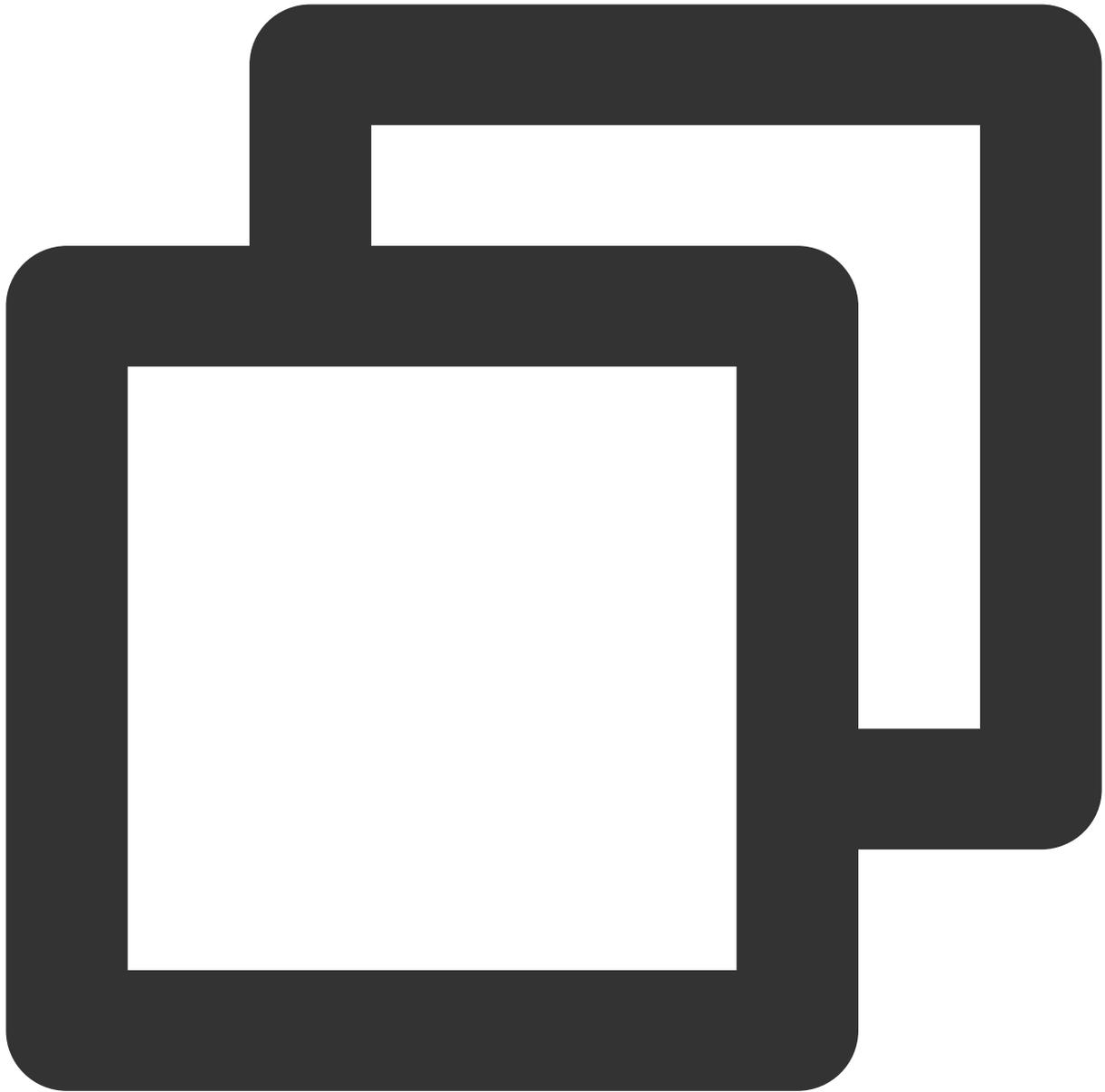
否，请 [联系我们](#)。

6. 执行以下命令，检查防盗链设置是否正确。



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

返回如下信息，即表示配置正确。



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

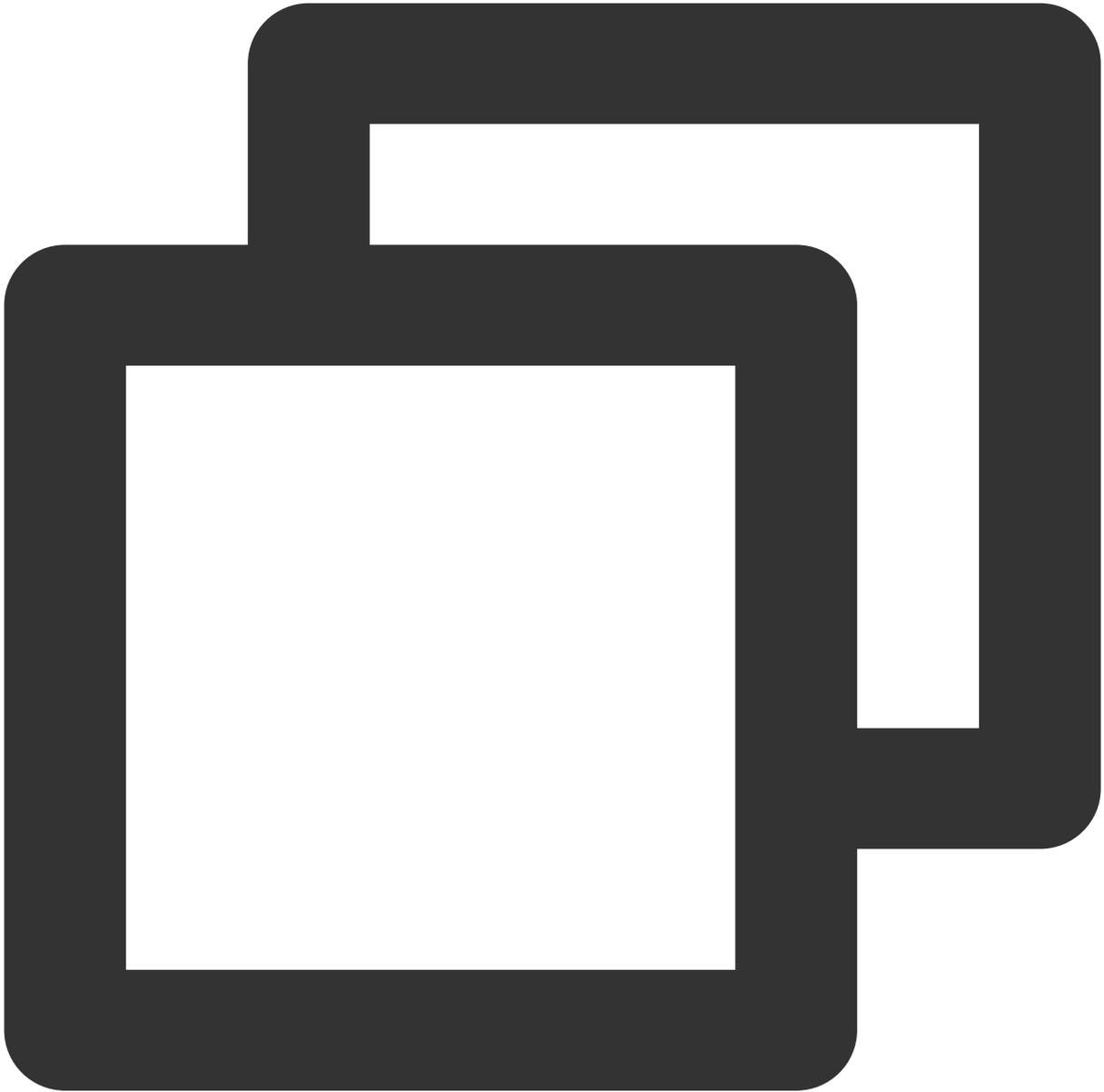
需要您执行以下操作：

1. 检查请求签名中 Authorization 的 q-ak 参数是否填写正确。
是，请执行下一步。
否，请修改 q-ak 参数。密钥的 SecretId 需与 q-ak 参数保持一致，且区分大小写。
2. 前往 [API密钥管理](#)，检查 API 密钥是否已启用。
是，请 [联系我们](#)。

否，请启用该 API 密钥。

Message 为 “InvalidObjectState”

当您访问 COS 出现如下信息时：



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

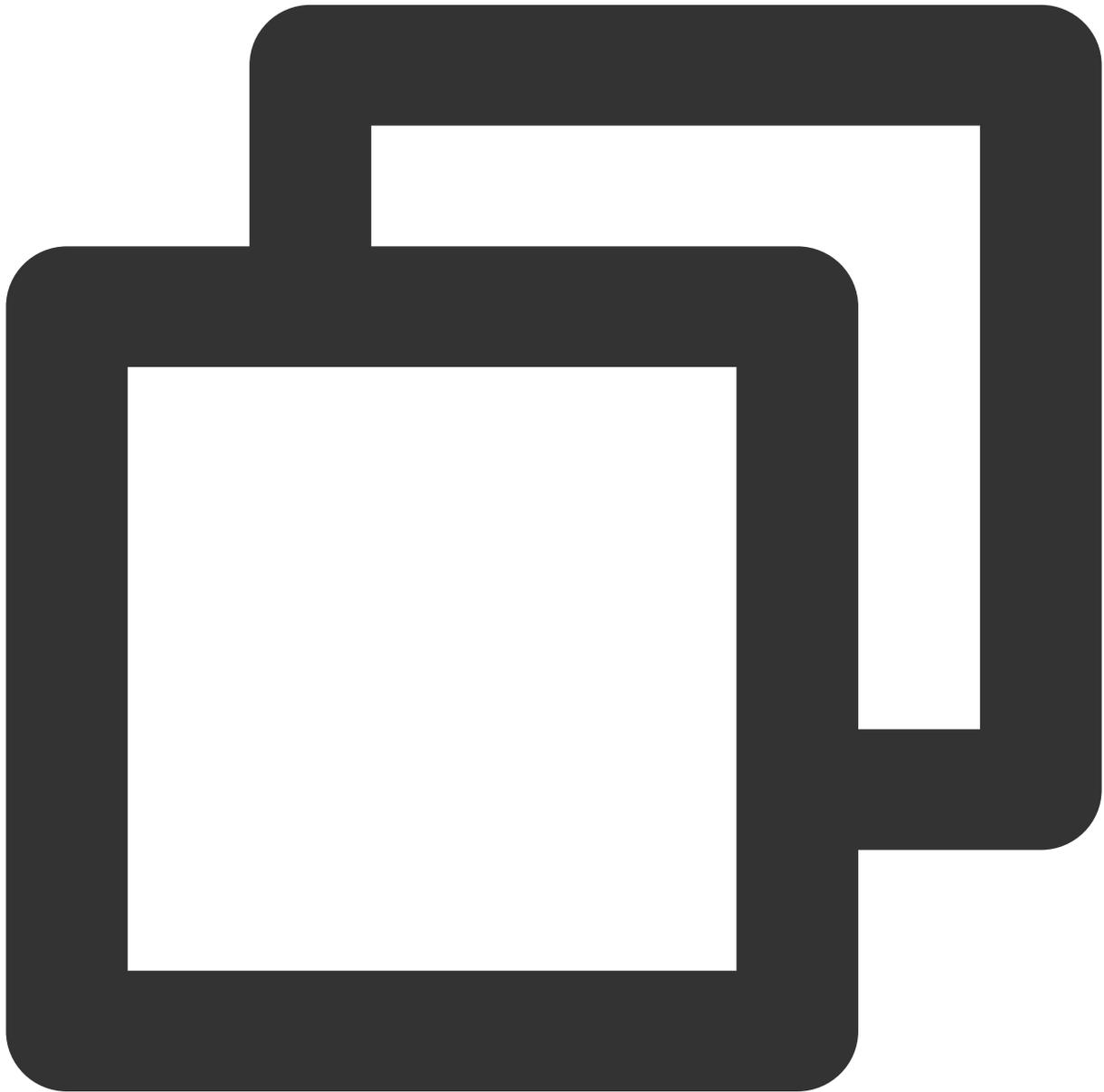
需要您执行以下操作：

检查请求的对象是否为归档类型或深度归档类型。

是，请恢复对象，再进行访问。详细操作请参见 [POST Object restore](#)。
否，请 [联系我们](#)。

Message 为 “RequestTimeTooSkewed”

当您访问 COS 出现如下信息时：



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

需要您执行以下操作：

1. 根据操作系统类型，查看客户端当前时间。

Windows 系统（以 Windwos Server 2012 为例）：



> 控制面板 > 时钟、语言和区域 > 设置日期和时间。

Linux 系统：执行 `date -R` 命令。

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

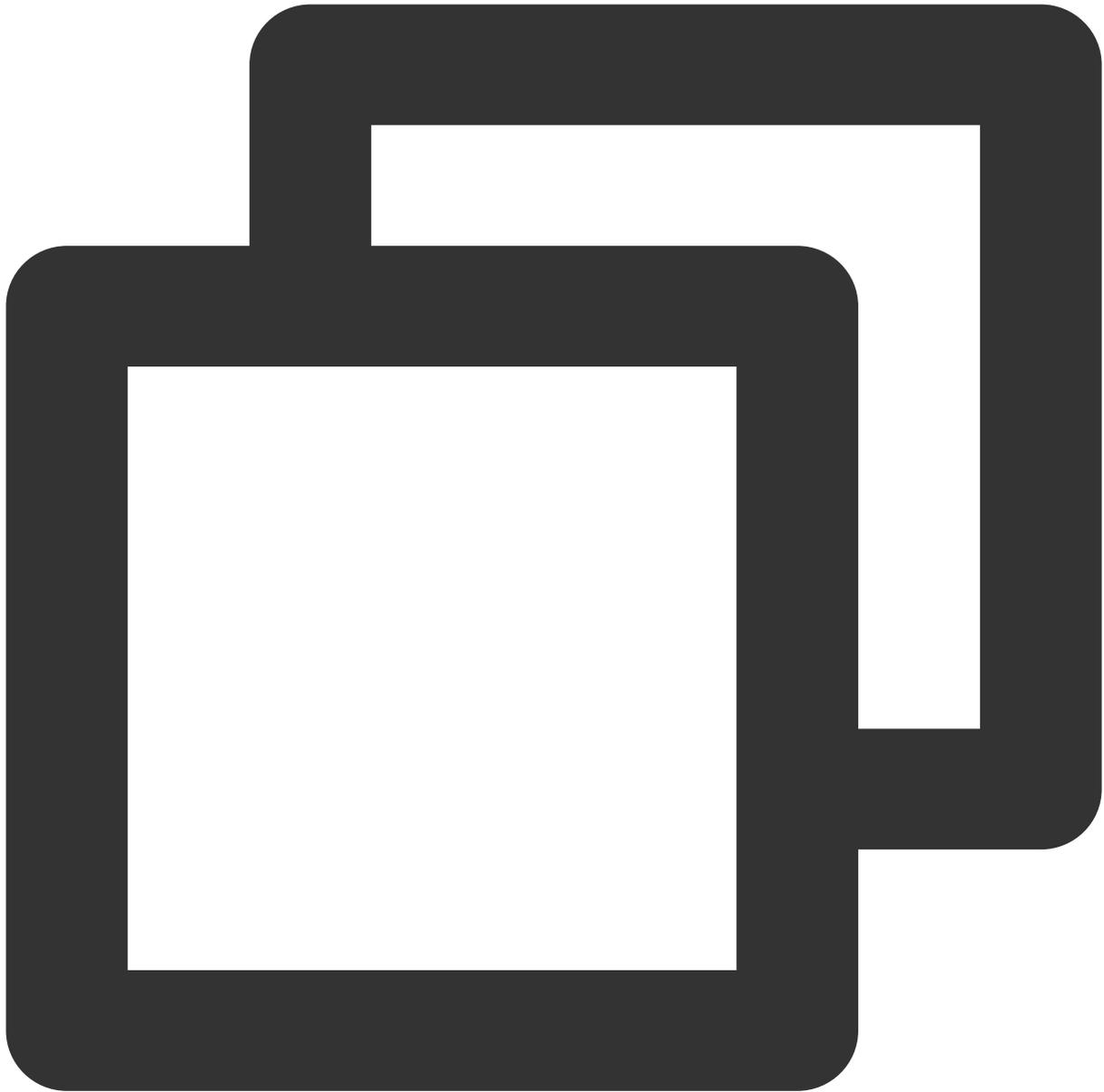
2. 判断客户端当前时间与服务器的时间是否存有偏差（时间偏差超过15分钟）。

是，请同步时间。

否，请 [联系我们](#)。

Message 为“Request has expired”

当您访问 COS 出现如下信息时：



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

产生的可能原因如下：

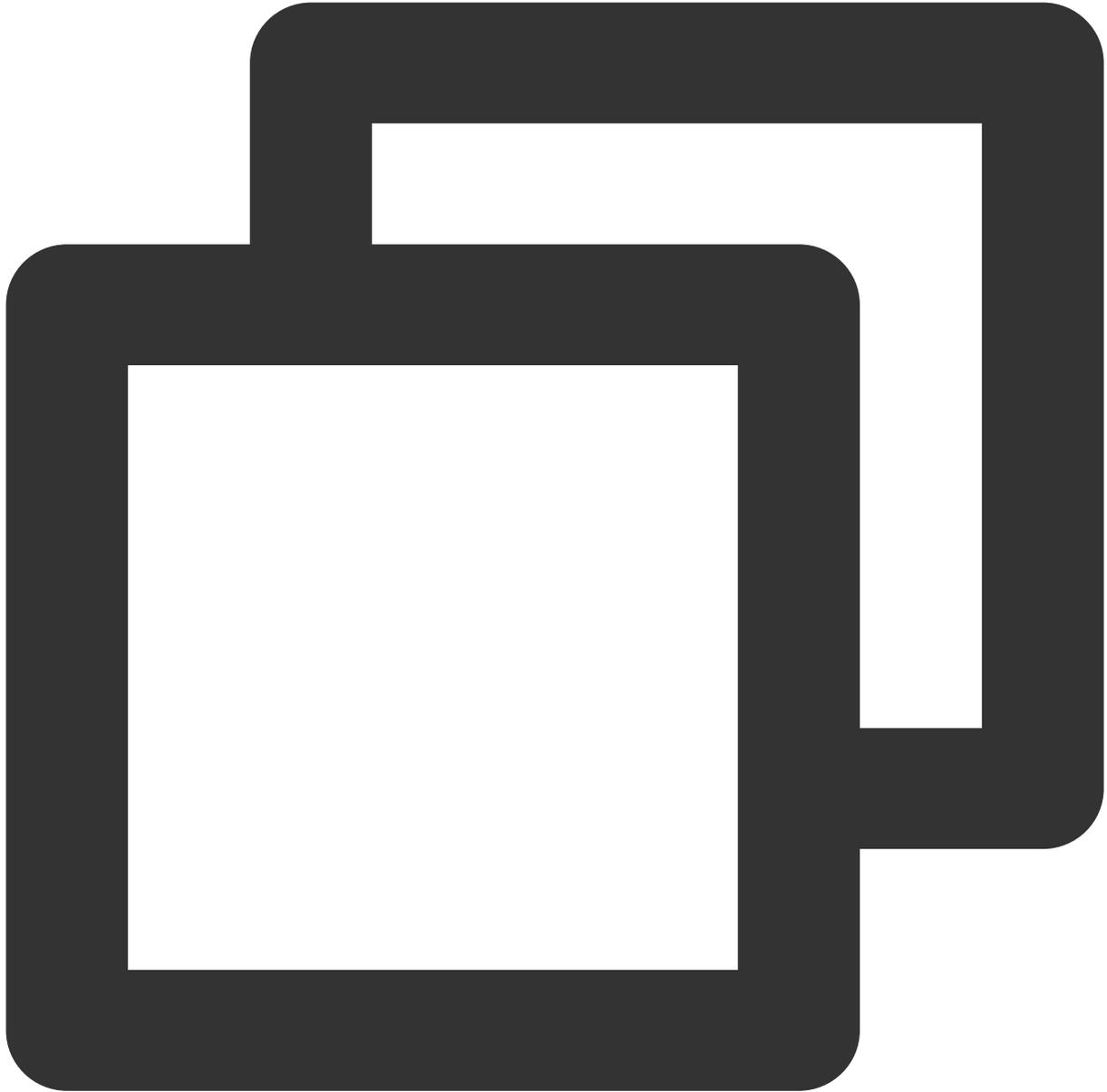
发起请求的时间超过了签名的有效时间。

本地系统时间和所在时区的时间不一致。

需要您重新设置签名的有效时间，或者同步本地系统时间。若仍无法解决，请 [联系我们](#)。

Message 为 “SignatureDoesNotMatch”

当您访问 COS 出现如下信息时：



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

需要您执行以下操作：

检查客户端计算的签名与服务端计算的签名是否一致。

是，请 [联系我们](#)。

否，请参阅 [请求签名](#) 文档，并使用COS 签名工具检查自行实现的签名过程。

访问视频时无法正常播放

最近更新时间：2024-01-06 11:27:57

现象描述

播放视频时，无法正常播放。例如，视频只有声音没有画面，只有画面没有声音，甚至无法播放等问题。

可能原因

原视频的编码格式有问题。

原视频被损坏，导致上传视频后存在播放时只有声音或只有画面，甚至无法播放等问题。

处理步骤

1. 检查原视频的编码格式是否正确。

是，请执行下一步。

否，请上传正确编码格式的视频。具体要求请参考 [规格与限制](#)。

2. 使用 VLC 等专业播放器检查原视频是否可以正常播放。

是，请 [联系我们](#)。

否，建议使用腾讯云的转码产品尝试修复原视频。详情请参考 [视频转码](#)。

同一个链接访问的文件异常

最近更新时间：2024-01-06 11:27:57

现象描述

现象一：访问同一个链接，但访问的文件不相同。

现象二：更新文件后，访问的文件还是旧文件。

可能原因

内容分发网络（Content Delivery Network，CDN）缓存没有过期。

浏览器没有禁用本地缓存。

访问的文件可能被劫持，导致访问的资源内容和预期的内容不一致。

处理步骤

检查 CDN 缓存是否过期

参考 [CDN 缓存配置](#) 常见问题文档，判断 CDN 缓存是否过期。

是，请 [检查浏览器是否禁用本地缓存](#)。

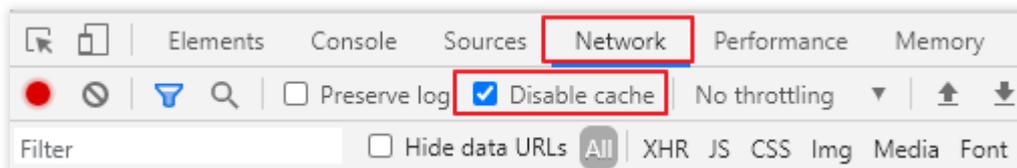
否，请参考 [缓存刷新](#) 文档，刷新 CDN URL 或 CDN 目录。

检查浏览器是否禁用本地缓存

说明

如下操作以 Google 浏览器为例。

1. 打开 Google 浏览器。
2. 按 **F12**，打开调试窗口。
3. 选择 **Network** 页签，查看是否勾选了 **Disable cache**。



是，请 [检查访问的文件是否被劫持](#)。

否，请勾选 **Disable cache**，并重启浏览器。

检查访问的文件是否被劫持

当您访问的资源内容和预期的不一致时（例如，文件的 content-length 不一样，响应的 header 不一致等），即表示被劫持。建议您使用 HTTPS 协议访问该文件。

使用 CDN 域名访问 COS 返回 HTTP ERROR 403

最近更新时间：2024-01-06 11:27:57

现象描述

使用内容分发网络（Content Delivery Network，CDN）域名访问对象存储（Cloud Object Storage，COS）时，返回 HTTP ERROR 403 错误码。

可能原因

CDN 加速域名为关闭状态。

处理步骤

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，选择**存储桶列表**，进入存储桶管理页面。
3. 找到需要操作的存储桶，单击该存储桶名称，进入存储桶配置页面。
4. 在左侧导航栏中，选择**域名与传输管理 > 默认 CDN 加速域名**，进入默认 CDN 加速域名页面。
5. 在“默认 CDN 加速域名”栏中，检查当前状态是否为关闭状态。
是，请 [开启默认 CDN 加速域名](#)。
否，请执行下一步。
6. 在“自定义 CDN 加速域名”栏中，检查状态是否为已上线。
是，请 [联系我们](#)。
否，请 [开启自定义 CDN 加速域名](#)。

使用 HTTPS 协议访问自定义源站域名报错

最近更新时间：2024-01-06 11:27:57

现象描述

使用 HTTPS 协议访问自定义源站域名时报错。

可能原因

证书配置错误或没有配置自定义源站域名。

处理步骤

使用 CDN 证书

1. 登录 [内容分发网络控制台](#)。
 2. 在左侧导航栏中，选择**域名管理**，进入域名管理页面。
 3. 单击需要配置的域名，选择 **HTTPS配置**页签，进入 HTTPS 配置页面。
 4. 在“HTTPS配置”栏中，单击**前往配置**，进行证书配置。
- 详情请参考内容分发网络的 [证书配置](#) 文档。
5. 等待约5分钟，待 CDN 域名重新完成部署后，即可成功访问 HTTPS 协议。

使用 COS 证书

1. 登录 [对象存储控制台](#)。
2. 在左侧导航栏中，选择**存储桶列表**，进入存储桶管理页面。
3. 找到需要操作的存储桶，单击该存储桶名称，进入存储桶配置页面。
4. 在左侧导航栏中，选择**域名与传输管理 > 自定义源站域名**，进入自定义源站域名页面。
5. 选择需要操作的域名，单击**绑定证书**，进行证书配置。
6. 单击**确定**，完成绑定。

当“HTTPS 证书”为“已上传”时，即可成功访问 HTTPS 协议。

使用 CVM 反代理证书

参考 [配置自定义域名支持 HTTPS 访问](#)。

POST Object 常见异常

最近更新时间：2024-01-06 11:27:57

故障现象

使用 COS API 进行 POST 请求时，返回如下异常错误码：

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]

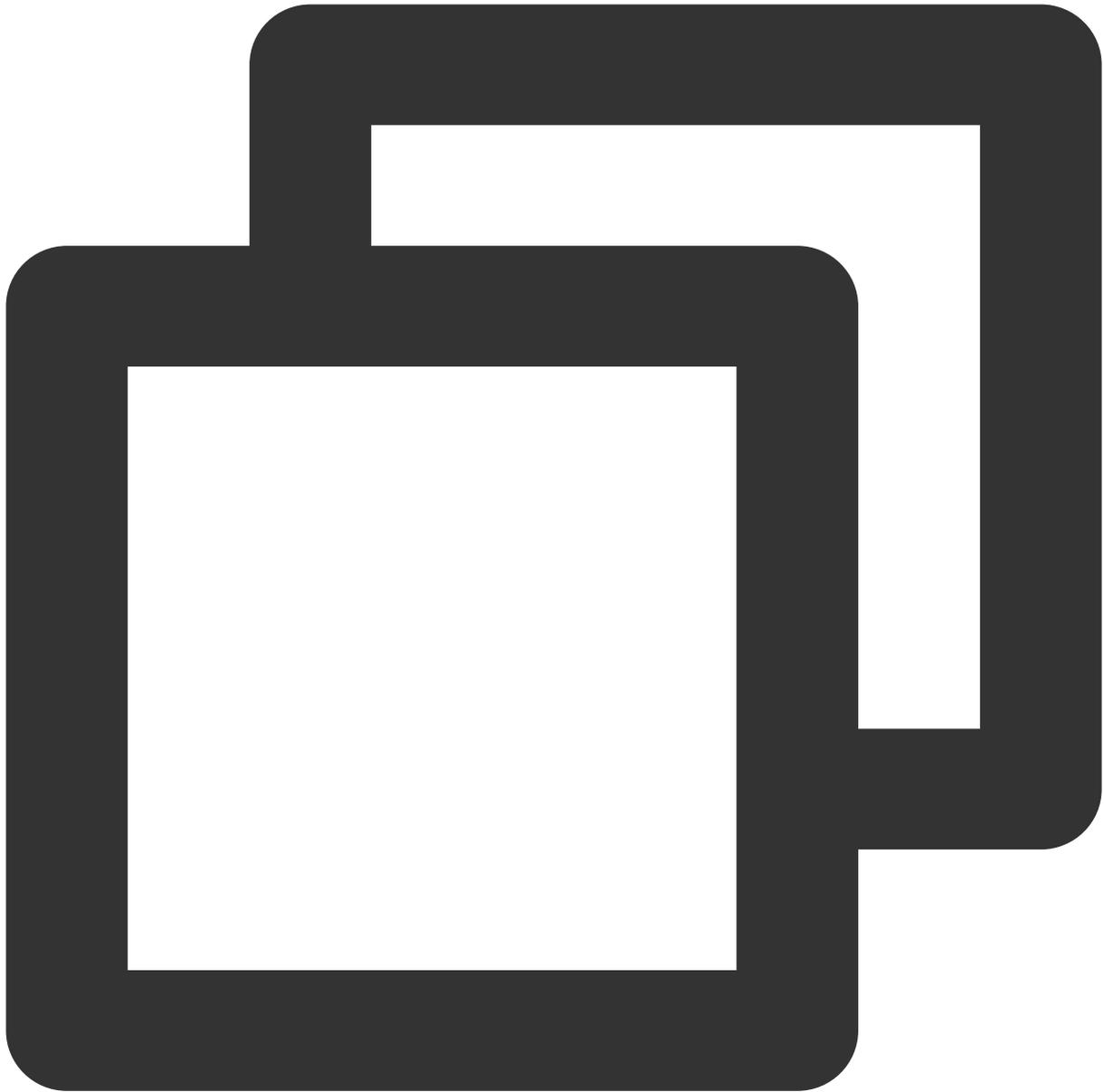
Condition key key doesn't match the value XXXXX

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

故障定位及处理

Message 为“Condition key q-ak doesn't match the value XXXXXX”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

可能原因

q-ak 参数输入错误。

解决方法

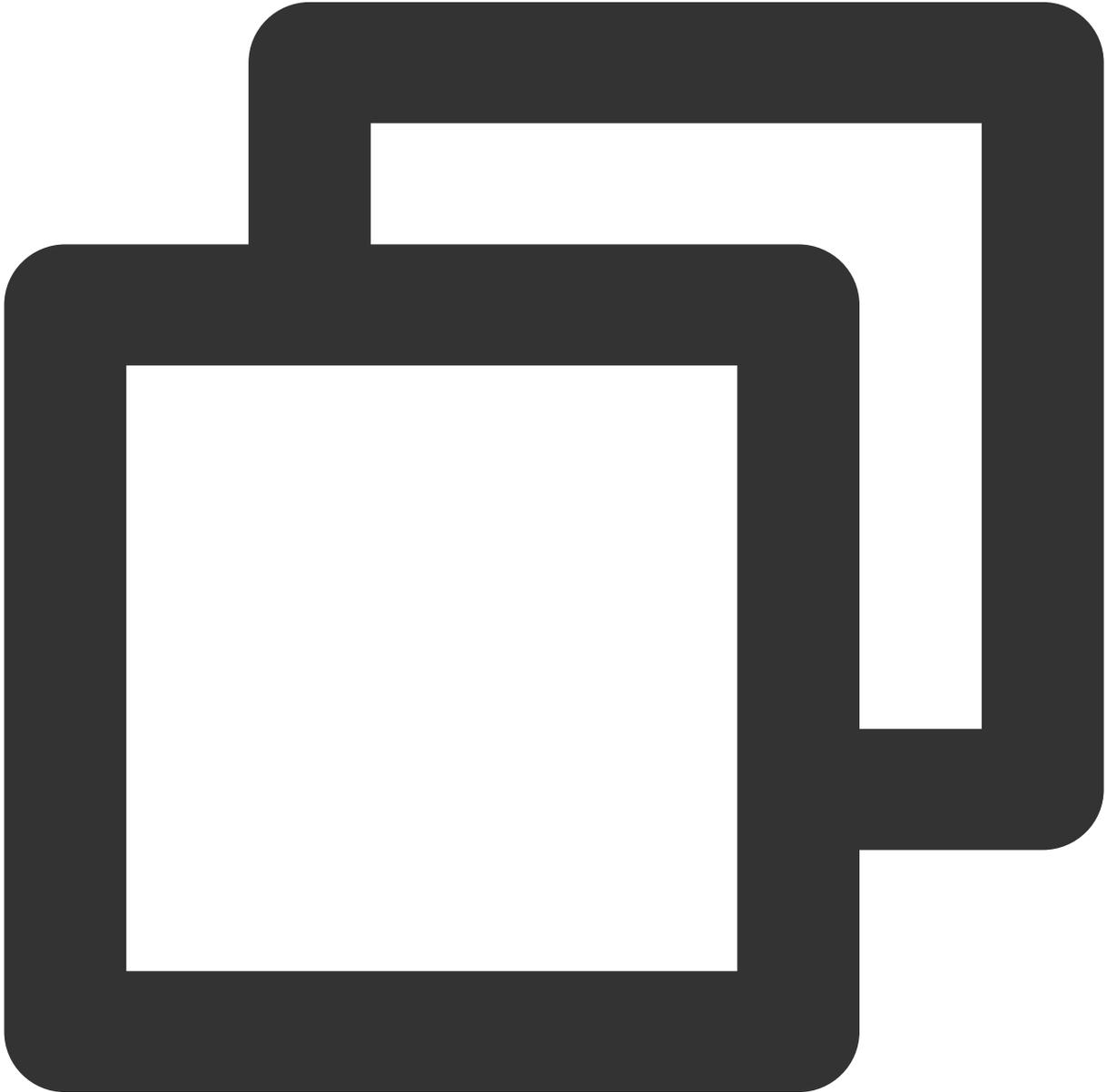
1. 登录访问管理控制台，进入 [API 密钥管理](#) 页面，查看密钥信息。
2. 根据查看的密钥信息，确认 q-ak 参数是否输入错误。

是，请将 q-ak 参数修改为正确的 SecretId。

否，请 [联系我们](#)。

Message 为 “You post object request has been expired, expiration time: 1621188104 but the time now : 1621245817”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

```
<Message>You post object request has been expired, expiration time: 1621188104 but
```

可能原因

Policy 中的 expiration 值已过期。

解决方法

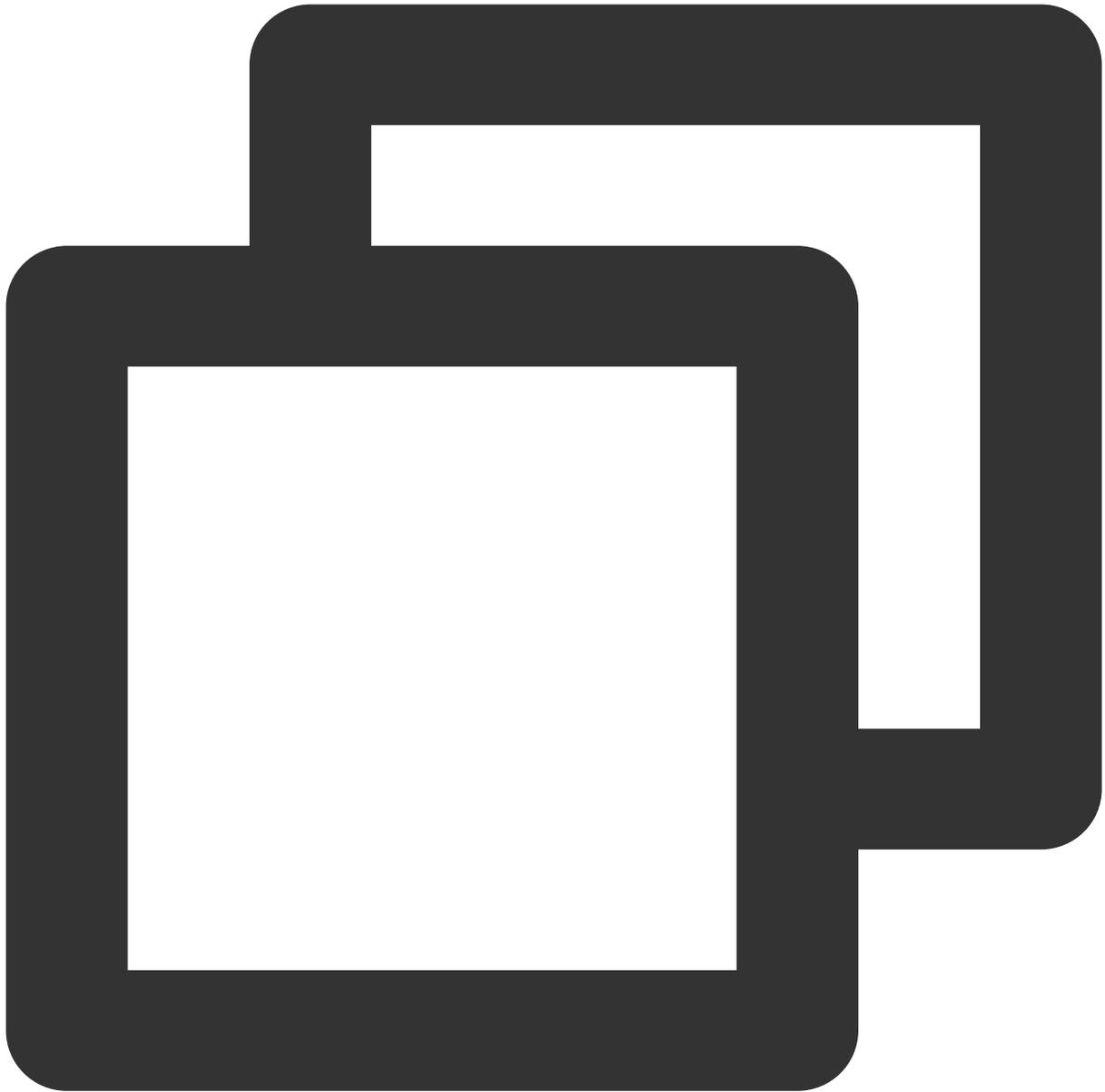
请修改 Policy 中的 expiration 值。

注意

expiration 值需要晚于当前时间，建议设置为当前时间+30分钟（UTC 时间）。

Message 为 “The Signature you specified is invalid.”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

可能原因

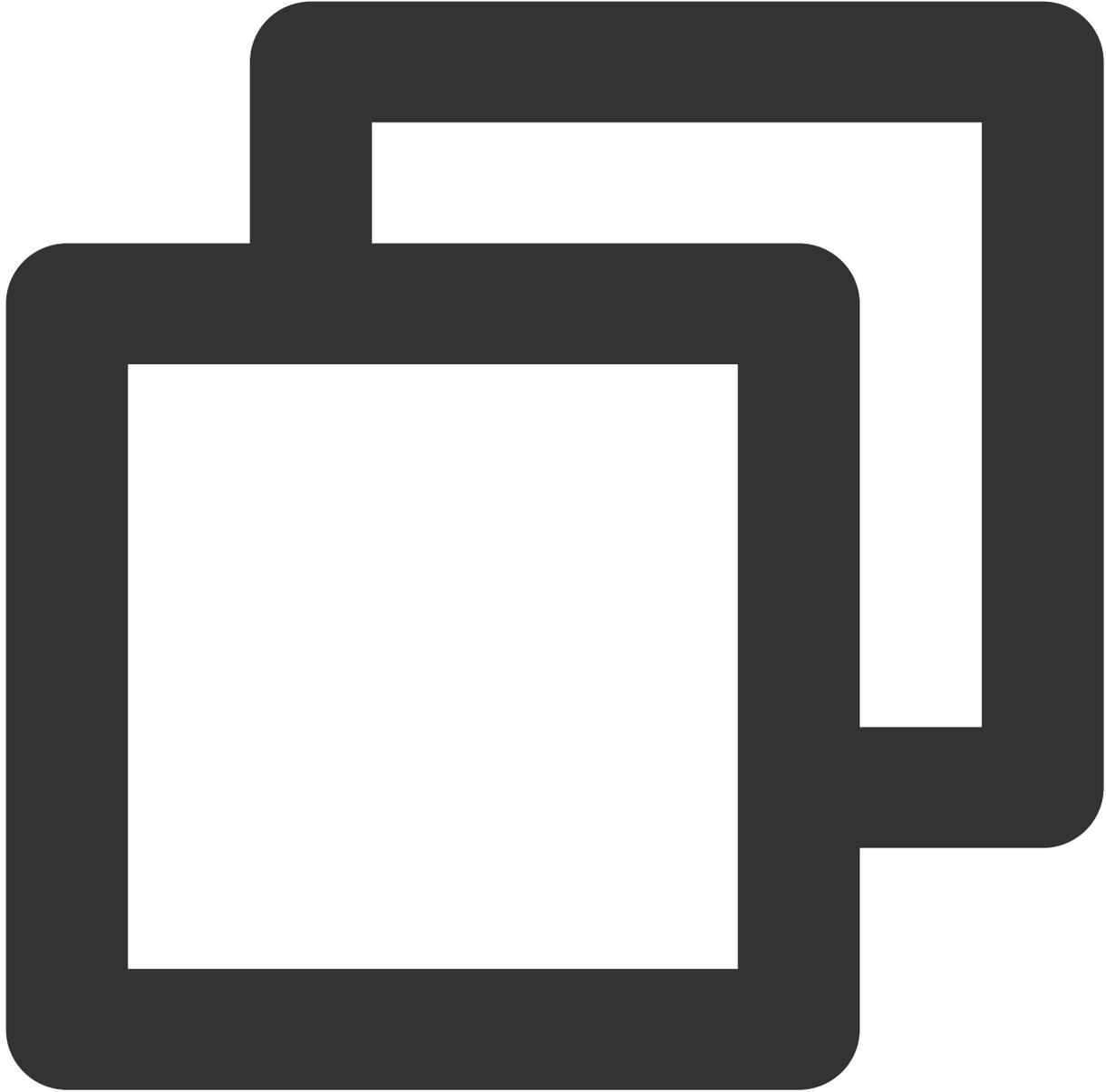
签名计算错误。

解决方法

请参考 [请求签名](#) 文档，检查 POST 签名串生成规则是否正确。

Message 为 “You must provide condition if you specify a policy in post object request.”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

可能原因

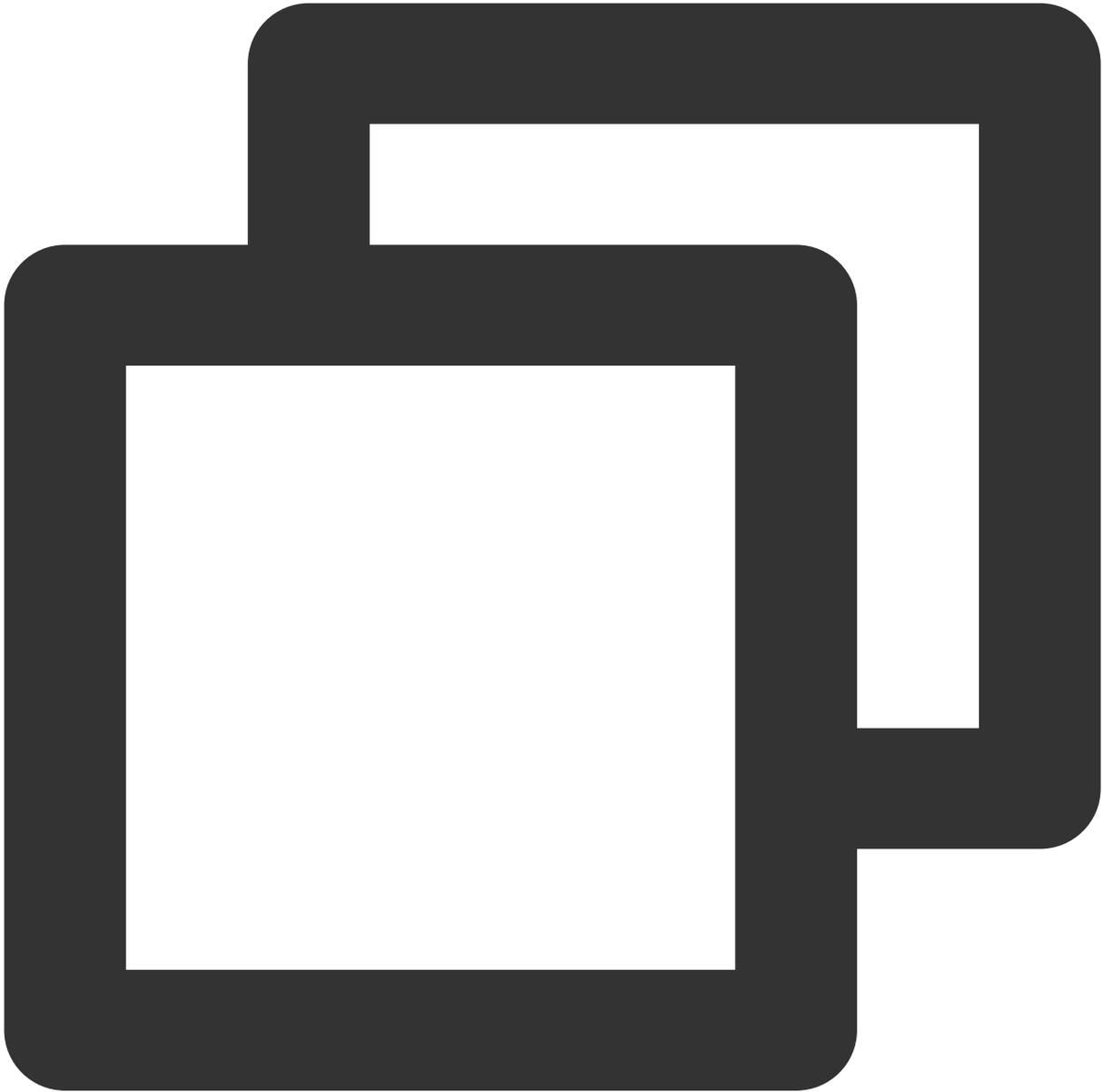
Policy 格式错误。

解决方法

请参考 [POST Object](#) 文档，将 Policy 格式修改为标准 JSON 格式。

Message 为 “Condition key bucket doesn't match the value [bucket-appid]”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

可能原因

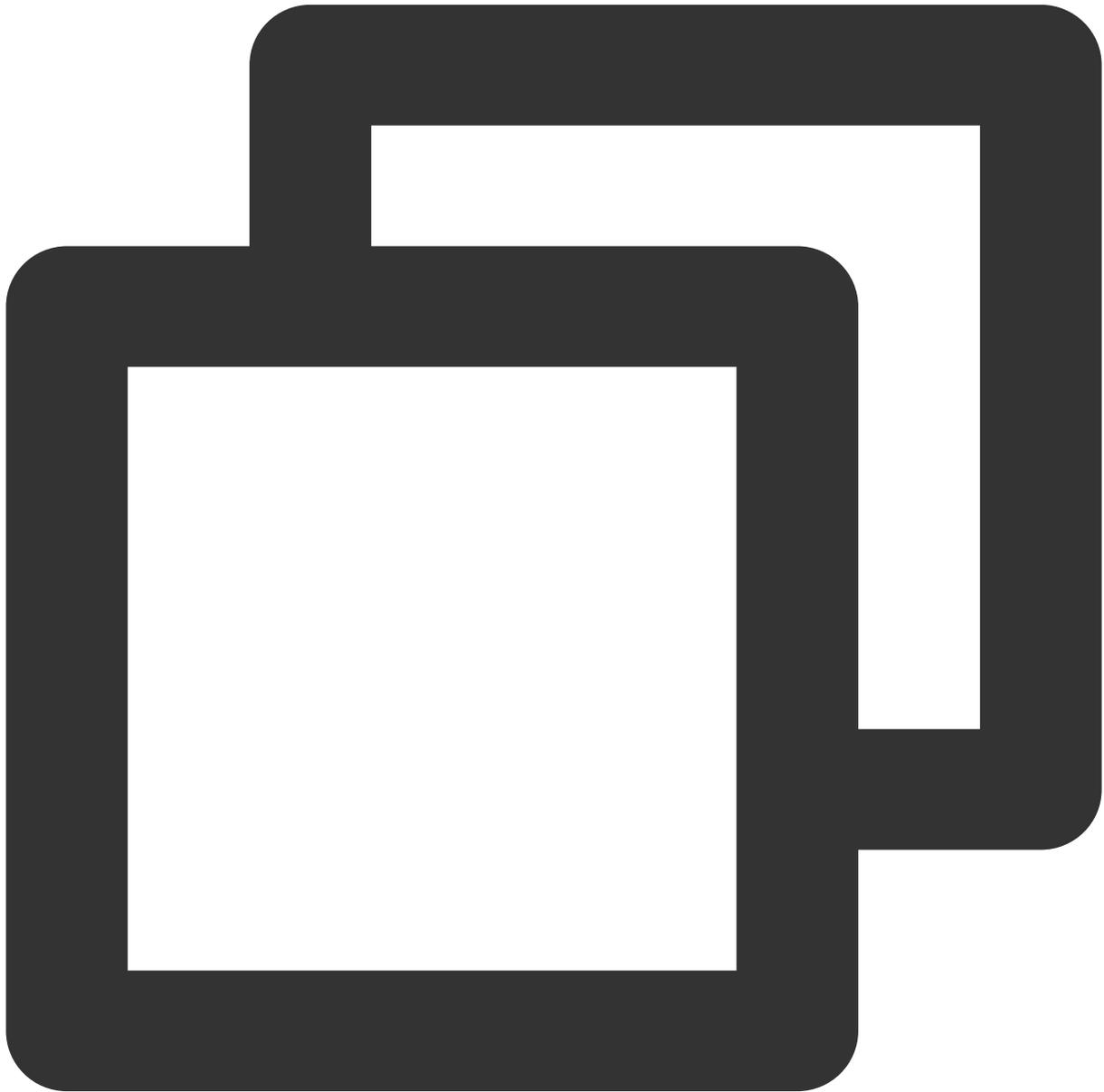
Policy 中的 bucket 与请求 bucket 不一致。

解决方法

请使用 Policy 中的 bucket 进行请求。

Message 为 “Condition key key doesn't match the value XXXXX”

当您使用 COS API 进行 POST 请求出现如下信息时：



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

可能原因

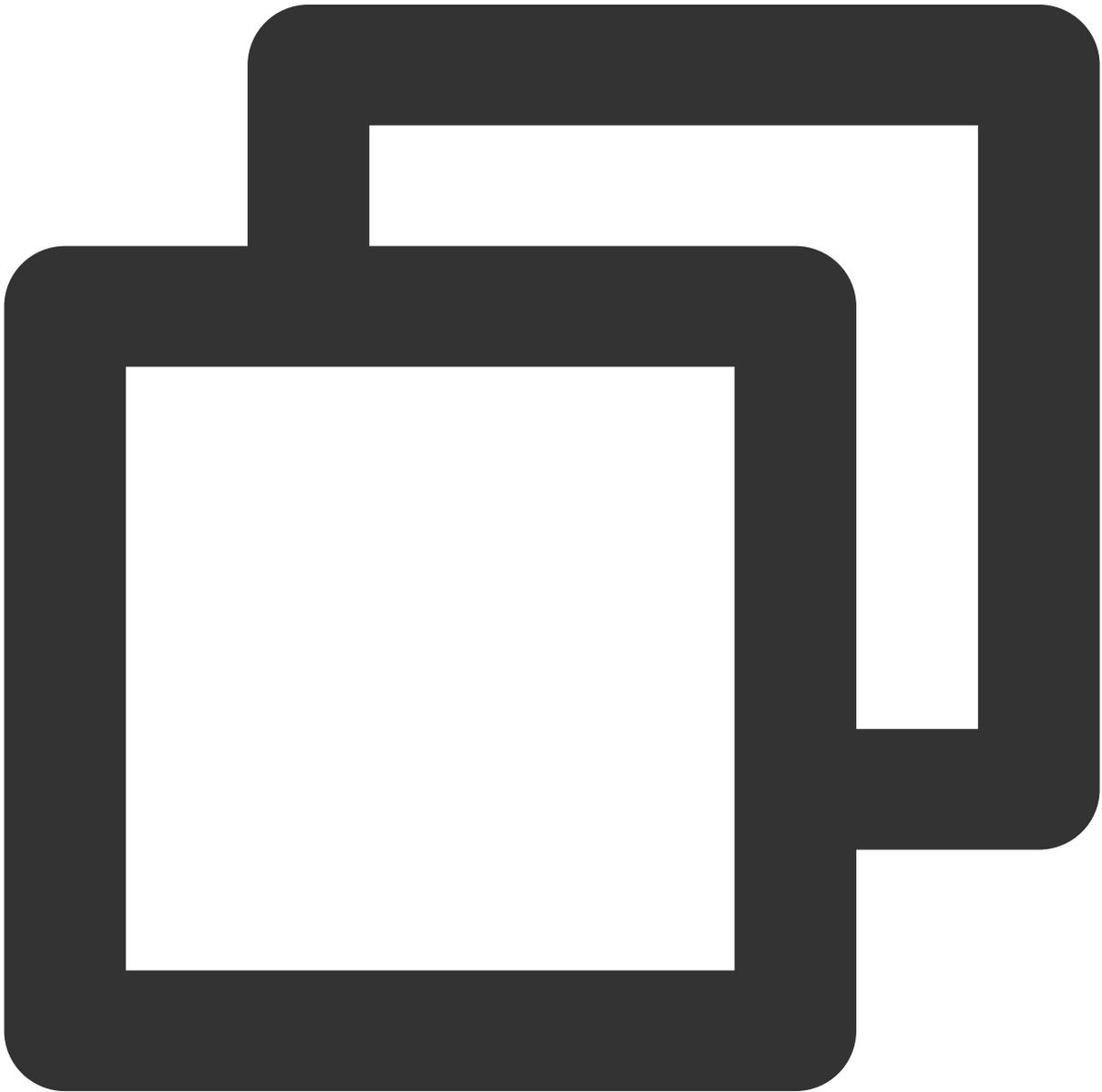
上传的内容不符合 policy 规则。

解决方法

根据 Policy 的 Condition，上传符合该条件的内容。

Message 为 “The body of your POST request is not well-formed multipart/form-data.”

当您使用 COS API 进行 POST 请求出现如下信息时：



```
<Code>MalformedPOSTRequest</Code>  
<Message>The body of your POST request is not well-formed multipart/form-data.</Mes
```

可能原因

POST body 格式不符合规范。

解决方法

参考 [POST Object](#) 文档，优化 body 格式。