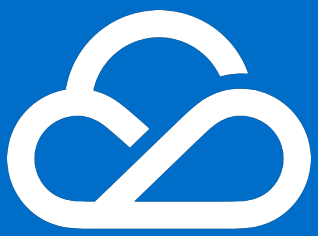
腾讯云对象存储

SDK 文档

产品文档



腾讯云

【版权声明】

【商标声明】

【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模式的承诺或保证。

文档目录

SDK 总览

SDK 概览

除了直接使用 API 接口外，COS 提供了丰富多样的 SDK 供开发者使用。

SDK	接入文档
PHP SDK	PHP SDK 接入说明
Python SDK	Python SDK 接入说明
Java SDK	Java SDK 接入说明
C++ SDK	C++ SDK 接入说明
C sharp SDK	C sharp SDK 接入说明
Android SDK	Android SDK 接入说明
iOS SDK	iOS SDK 接入说明

Android SDK

开发准备

SDK 获取

对象存储服务的 Android SDK 的下载github地址:Android SDK。

更多示例可参考Demo:Android SDK Demo。

开发准备

1. SDK 支持 Android 2.2 及以上版本的手机系统；
2. 手机必须要有网络（GPRS、3G或 WIFI 网络等）；
3. 手机可以没有存储空间，但会使部分功能无法正常工作；
4. 从控制台获取APP ID、SecretID、SecretKey，详情参考权限控制。

SDK 配置

配置工程导入下列 jar 包：

- cos-android-sdk1.4.2.jar
- okhttp-3.2.0.jar
- okio-1.6.0.jar

SDK 需要网络访问相关的一些权限，需要在 AndroidManifest.xml 中增加如下权限声明：

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

初始化

进行操作之前需要实例化COSClient 和 COSClientConfig；

实例化 COSClientConfig

调用 COSClientConfig() 方法实例化 COSClientConfig 对象。

```
COSClientConfig config = new COSClientConfig();
```

配置 COSClientConfig

方法	方法描述
setEndPoint(COSEndPoint endPoint)	设置园区：华南 COSEndPoint.COS_GZ; 华北 COSEndPoint.COS_TJ ;sdk中默认为华南地区
setConnectionTimeout(int connectionTimeout)	连接超时设置
setSocketTimeout(int socketTimeout)	读取超时设置
setMaxConnectionsCount(int maxConnectionsCount)	并发数大小设置
setMaxRetryCount(int maxRetryCount)	失败请求重试次数

实例化 COSClient

调用 COSClient（Context context, String appid, COSClientConfig config, String peristenceId) 方法实例化 COSClient 对象。

参数说明

参数名称	类型	是否必填	参数描述
context	Context	是	上下文
appid	String	是	腾讯云注册的APPID
config	COSClientConfig	否	配置设置
persistenceId	String	否	持久化 ID，每个 COSClient 需设置一个唯一的 ID 用于持久化保存未完成任务列表，以便应用退出重进后能够继续进行上传；传入为 Null，则不会进行持久化保存

示例

```
//创建COSClientConfig对象，根据需要修改默认的配置参数
COSClientConfig config = new COSClientConfig();
//设置园区
config.setEndPoint(COSEndPoint.COS_GZ);

Context context = getApplicationContext();
String appid =  "腾讯云注册的appid";
String peristenceId = "持久化Id";

//创建COSlient对象，实现对象存储的操作
COSClient cos = new COSClient(context,appid,config,peristenceId);
```

快速入门

初始化 COSClient

```
String appid =  "腾讯云注册的appid";
Context context = getApplicationContext();
String peristenceId = "持久化Id";

//创建COSClientConfig对象，根据需要修改默认的配置参数
COSClientConfig config = new COSClientConfig();
//如设置园区
config.setEndPoint(COSEndPoint.COS_GZ);

COSClient cos = new COSClient(context,appid,config,peristenceId);
```

上传文件

```
String bucket = "cos空间名称";
String cosPath = "远端路径，即存储到cos上的路径";
String srcPath = "本地文件的绝对路径";
String sign = "签名，此处使用多次签名";

PutObjectRequest putObjectRequest = new PutObjectRequest();
putObjectRequest.setBucket(bucket);
putObjectRequest.setCosPath(cosPath);
putObjectRequest.setSrcPath(srcPath);
putObjectRequest.setSign(sign);
putObjectRequest.setListener(new  IUploadTaskListener(){
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {

        PutObjectResult result = (PutObjectResult) cosResult;
```

```
            if(result != null){
                StringBuilder stringBuilder = new StringBuilder();
                stringBuilder.append(" □□□□□ ret="
    + result.code + "; msg =" +result.msg + "\n");
                stringBuilder.append(" access_url= "
    + result.access_url == null ? "null" :result.access_url + "\n");
                stringBuilder.append(" resource_path= "
    + result.resource_path == null ? "null" :result.resource_path + "\n");
                stringBuilder.append(" url= " + result.url == null ? "null"
    :result.url);
                Log.w("TEST",stringBuilder.toString());
             }
        }

        @Override
        public void onFailed(COSRequest COSRequest, final COSResult cosResult) {
            Log.w("TEST","□□□□□ ret =" +cosResult.code + "; msg ="
    + cosResult.msg);
        }

        @Override
        public void onProgress(COSRequest cosRequest, final long
    currentSize, final long totalSize) {
            float progress = (float)currentSize/totalSize;
            progress = progress *100;
            Log.w("TEST","□□□  " + (int)progress + "%");
         }
});
```

下载文件

```
String downloadURl =  "下载文件的URL";
String savePath = "本地保存文件的路径";
String sign = "开启token防盗链了，则需要签名；否则，不需要";

GetObjectRequest getObjectRequest = new GetObjectRequest(downloadURl,savePath);
getObjectRequest.setSign(null);
getObjectRequest.setListener(new IDownloadTaskListener() {
    @Override
    public void onProgress(COSRequest cosRequest, final long
 currentSize, final long totalSize) {
        float progress =  currentSize / (float)totalSize;
        progress = progress * 100;
        progressText.setText("progress =" + (int) (progress) + "%");
        Log.w("TEST", "progress =" + (int) (progress) + "%");
   }

   @Override
   public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
       Log.w("TEST","code =" + cosResult.code + "; msg =" + cosResult.msg);
   }

   @Override
   public void onFailed(COSRequest COSRequest, COSResult cosResult) {
        Log.w("TEST","code =" + cosResult.code + "; msg =" + cosResult.msg);
   }
});

GetObjectResult getObjectResult = cos.getObject(getObjectRequest);
```

生成签名

签名类型：

类型	含义
多次有效	有效时间内多次始终有效
单次有效	与资源URL绑定，一次有效

签名获取：

SDK 中用到的 SIGN，推荐使用 服务器端SDK，并由移动端向业务服务器请求。SIGN 的具体生成和使用请参照 访问权限。

目录操作

创建目录

方法原型

调用此接口可在指定的 bucket 下创建目录，具体步骤如下：

1. 调用 CreateDirRequest() 实例化 CreateDirRequest 对象；
2. 调用 COSClient 的 createDir 方法，传入 CreateDirRequest，返回 CreateDirResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	需要创建目录的路径
biz_attr	String	否	目录绑定的属性信息，由业务维护
sign	String	是	签名信息，此处使用多次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过CreateDirResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息
ctime	long(Unix时间戳)	创建时间

示例

```
CreateDirRequest createDirRequest = new CreateDirRequest();
createDirRequest.setBucket(bucket);
createDirRequest.setCosPath(cosPath);
createDirRequest.setBiz_attr(biz_attr);
createDirRequest.setSign(sign);
createDirRequest.setListener(new ICmdTaskListener() {
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
        final CreateDirResult createDirResult = (CreateDirResult) cosResult;
        Log.w("TEST","□□□□□□□ ret="
 + createDirResult.code + "; msg=" + createDirResult.msg
            +"ctime = " + createDirResult.ctime));
    }

    @Override
    public void onFailed(COSRequest COSRequest, final COSResult cosResult) {
         Log.w("TEST","□□□□□□□ ret=" + cosResult.code + "; msg="
 + cosResult.msg);
    }
});

CreateDirResult result = cos.createDir(createDirRequest);
```

目录列表查询

方法原型

调用此接口可查询指定目录下的列表信息，具体步骤如下：

1. 调用 ListDirRequest() 实例化 ListDirRequest 对象；
2. 调用 COSClient 的 listDir 方法，传入 ListDirRequest，返回 ListDirResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
num	int	否	返回的数目，默认为1000，最大1000
content	String	否	透传字段，首次拉取必须清空。拉取下一页，需要将前一页返回值中的context透传到参数中
prefix	String	否	前缀查询的字符串
sign	String	是	签名信息，此处使用多次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过ListDirResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息
content	String	透传字段
listover	boolean	标识是否还有数据， true：列举结束；false：还有数据
infos	List	列举目录列表中文件或文件夹的属性

示例

```
ListDirRequest listDirRequest = new ListDirRequest();
listDirRequest.setBucket(bucket);
listDirRequest.setCosPath(cosPath);
listDirRequest.setNum(100);
listDirRequest.setContent("");
listDirRequest.setSign(sign);
listDirRequest.setListener(new ICmdTaskListener() {
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
            //查询成功
        ListDirResult listObjectResult = (ListDirResult) cosResult;
        if(listObjectResult.infos != null && listObjectResult.infos.size() > 0) {
        for(int i = 0; i < length; i++){
            String str = listObjectResult.infos.get(i);
            try {
                JSONObject jsonObject = new JSONObject(str);
                if(jsonObject.has("sha")){
                 //当前搜索的结果是文件
                }else{
                //当前搜索的结果是文件夹
                }
            } catch (JSONException e) {
               e.printStackTrace();
           }
     }
     }

        if (!listover) {
            //  存在下一页保存当前页的状态信息
            String content = result.content;
        }
    }
```

```
    @Override
    public void onFailed(COSRequest COSRequest, final COSResult cosResult) {
        Log.w("TEST",□□□□□□□ret=" + cosResult.code  + "; msg =
" + cosResult.msg);
    }
});


// □□□□□□
listDirRequest.setPrefix(prefix);
ListDirResult result=cos.listDir(listDirRequest);
```

目录更新

方法原型

调用此接口可更新指定目录的信息，具体步骤如下：

1. 调用 UpdateObjectRequest() 实例化 UpdateObjectRequest 对象；
2. 调用 COSClient 的 updateObject 方法，传入 UpdateObjectRequest，返回 UpdateObjectResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处使用单次签名
bizAttr	String	否	目录绑定的属性信息
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过UpdateObjectResult对象的成员变量成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息

示例

```
UpdateObjectRequest updateObjectRequest = new UpdateObjectRequest();
updateObjectRequest.setBucket(bucket);
updateObjectRequest.setCosPath(cosPath);
updateObjectRequest.setBizAttr(biz_attr);
updateObjectRequest.setSign(onceSign);
updateObjectRequest.setListener(new ICmdTaskListener() {
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
        //□□□□
        Log.w("TEST", cosResult.code+" : "+cosResult.msg);
    }

     @Override
     public void onFailed(COSRequest COSRequest, COSResult cosResult) {
         //□□□□
         Log.w("TEST", cosResult.code+" : "+cosResult.msg);
     }
});

UpdateObjectResult result = cos.updateObject(updateObjectRequest);
```

目录查询

方法原型

调用此接口可查询指定目录的信息，具体步骤如下：

1. 调用 GetObjectMetadataRequest() 方法实例化 GetObjectMetadataRequest 对象；
2. 调用 COSClient 的 getObjectMetadata方法，传入 GetObjectMetadataRequest，返回 GetObjectMetadataResult对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处使用多次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过GetObjectMetadataResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息
biz_attr	String	目录绑定的属性信息
ctime	long(Unix时间戳)	创建时间
mtime	long(Unix时间戳)	最后一次修改时间

示例

```
GetObjectMetadataRequest get
ObjectMetadataRequest = new GetObjectMetadataRequest();
getObjectMetadataRequest.setBucket(bucket);
getObjectMetadataRequest.setCosPath(cosPath);
getObjectMetadataRequest.setSign(sign);
```

```
getObjectMetadataRequest.setListener(new ICmdTaskListener() {
     @Override
     public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
         GetObjectMetadataResult result = (GetObjectMetadataResult) cosResult;
         StringBuilder stringBuilder = new StringBuilder();
         stringBuilder.append("code=" + result.code + "; msg="
 +result.msg + "\n");
         stringBuilder.append("ctime =" +result.ctime + "; mtime="
 +result.mtime + "\n" );
         stringBuilder.append("biz_attr="
 + result.biz_attr == null ? "" : result.biz_attr );
         Log.w("TEST",stringBuilder.toString());
      }

      @Override
      public void onFailed(COSRequest cosRequest, final COSResult cosResult) {
          Log.w("TEST", cosResult.code+" : "+cosResult.msg);
       }

});

GetObjectMetadataRequest result = cos.getObjectMetadata(getObjectMetadataRequest
);
```

目录删除

方法原型

调用此接口可删除指定目录，具体步骤如下，注意只能删除空目录：

1. 调用 RemoveEmptyDirRequest() 方法实例化 RemoveEmptyDirRequest 对象；
2. 调用 COSClient 的 removeEmptyDir 方法，传入 RemoveEmptyDirRequest，返回

RemoveEmptyDirResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处使用单次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过RemoveEmptyDirResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息

示例

```
RemoveEmptyDirRequest removeEmptyDirRequest = new RemoveEmptyDirRequest();
removeEmptyDirRequest.setBucket(bucket);
removeEmptyDirRequest.setCosPath(cosPath);
removeEmptyDirRequest.setSign(onceSign);
removeEmptyDirRequest.setListener(new ICmdTaskListener() {
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
        final
 DeleteObjectResult removeEmptyDirResult = (DeleteObjectResult) cosResult;
        Log.w("TEST","removeDir □□□ ret="
 + removeEmptyDirResult.code + "; msg=" + removeEmptyDirResult.msg );
    }

    @Override
```

```
    public void onFailed(COSRequest COSRequest, final COSResult cosResult) {
        Log.w("TEST","□□□□□□□ret=" + cosResult.code  + "; msg ="
 + cosResult.msg);
    }
});

RemoveEmptyDirResult result = cos.removeEmptyDir(removeEmptyDirRequest);
```

文件操作

文件上传

方法原型

调用此接口上传指定的文件，具体步骤如下：

1. 调用 PutObjectRequest() 方法实例化 PutObjectRequest对象；
2. 调用 COSClient 的 putObject 方法，传入 PutObjectRequest，返回 PutObjectResult对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
srcPath	String	是	本地绝对路径
insertOnly	String	否	是否覆盖相同文件名："0"，允许覆盖；"1",不允许覆盖；默认为："1"．
slice_size	int	否	分片上传时，设置的分片大小，默认为:1M
sign	String	是	签名信息，此处使用多次签名

参数名称	类型	是否必填	参数描述
listener	IUploadTaskListener	否	结果回调

返回结果说明

通过PutObjectResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息
access_url	String	访问文件的URL
url	String	操作文件的url

示例

```
PutObjectRequest putObjectRequest = new PutObjectRequest();
putObjectRequest.setBucket(bucket);
putObjectRequest.setCosPath(cosPath);
putObjectRequest.setSrcPath(srcPath);
putObjectRequest.setSign(sign);

 /*  设置是否允许覆盖同名文件:  "0"，允许覆盖；"1",不允许覆盖；
putObjectRequest.setInsertOnly("1");

//设置是否开启分片上传
putObjectRequest.setSliceFlag(true);//设置是否分片上传: true，分片上传；false,简单文件上传
putObjectRequest.setSlice_size(1024*1024);//分片上传时，分片的大小
 */

putObjectRequest.setListener(new  IUploadTaskListener(){
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {

        PutObjectResult result = (PutObjectResult) cosResult;
        StringBuilder stringBuilder = new StringBuilder();
```

```
        stringBuilder.append(" □□□□□ ret=" + result.code + "; msg ="
 +result.msg + "\n");
        stringBuilder.append(" access_url= " + result.access_url + "\n");
        stringBuilder.append(" resource_path= " + result.resource_path + "\n");
        stringBuilder.append(" url= " result.url);
        Log.w("TEST",stringBuilder.toString();
     }

    @Override
    public void onFailed(COSRequest COSRequest, final COSResult cosResult) {
        Log.w("TEST","□□□□□ ret =" +cosResult.code + "; msg ="
 + cosResult.msg);
    }

    @Override
    public void onProgress(COSRequest cosRequest, final long
 currentSize, final long totalSize) {
        float progress = (float)currentSize/totalSize;
        progress = progress *100;
        Log.w("TEST","□□□  " + (int)progress + "%");
    }
});

PutObjectResult result = cos.putObject(putObjectRequest);
```

文件更新

方法原型

调用此接口可更新指定文件信息，具体步骤如下：

1. 调用 UpdateObjectRequest() 方法实例化 UpdateObjectRequest 对象；

2. 调用 COSClient 的 updateObject 方法，传入 UpdateObjectRequest，返回 UpdateObjectResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处使用单次签名
bizAttr	String	否	目录绑定的属性信息
authority	String	否	文件操作权限，与bucket拥有相同的权限(COSAuthority.EINVALID),私有读写权限（COSAuthority.EWRPRIVATE），私有写公有读权限（COSAuthority.EWPRIVATERPUBLIC）
custom_headers	Map	否	文件自定义的header: 如Cache-Control, Content-Disposition,Content-Language,x-cos-meta-。
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过UpdateObjectResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息

示例

```
UpdateObjectRequest updateObjectRequest = new UpdateObjectRequest();
```

```
updateObjectRequest.setBucket(bucket);
updateObjectRequest.setCosPath(cosPath);
updateObjectRequest.setBizAttr(biz_attr);
updateObjectRequest.setAuthority(authority);
updateObjectRequest.setCustomHeader(customer);
updateObjectRequest.setSign(onceSign);
updateObjectRequest.setListener(new ICmdTaskListener() {
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
        Log.w("TEST", cosResult.code+" : "+cosResult.msg);
    }

    @Override
    public void onFailed(COSRequest COSRequest, COSResult cosResult) {
                        Log.w("TEST", cosResult.code+" : "+cosResult.msg);
    }
});

UpdateObjectResult result= cos.updateObject(updateObjectRequest);
```

文件查询

方法原型

调用此接口可查询指定文件信息，具体步骤如下：

1. 调用 GetObjectMetadataRequest() 方法实例化 GetObjectMetadataRequest 对象；
2. 调用 COSClien t的 getObjectMetadata 方法，传入 GetObjectMetadataRequest，返回 GetObjectMetadataResult 对象；

参数说明

参数名称	类型	是否必填	参数描述

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处使用多次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过GetObjectMetadataResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息
biz_attr	String	目录绑定的属性信息
ctime	long(Unix时间戳)	创建时间
mtime	long(Unix时间戳)	最后一次修改时间
sha	String	文件的sha值
customs_headers	Map	文件的头部属性
filelen	int	文件的长度大小

示例

```
GetObjectMetadataRequest get
ObjectMetadataRequest = new GetObjectMetadataRequest();
getObjectMetadataRequest.setBucket(bucket);
getObjectMetadataRequest.setCosPath(cosPath);
getObjectMetadataRequest.setSign(sign);
getObjectMetadataRequest.setListener(new ICmdTaskListener() {
     @Override
     public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
         GetObjectMetadataResult result = (GetObjectMetadataResult) cosResult;
         StringBuilder stringBuilder = new StringBuilder();
         stringBuilder.append("code=" + result.code + "; msg="
```

```
+result.msg + "\n");
        stringBuilder.append("ctime =" +result.ctime + "; mtime="
+result.mtime + "\n" );
        stringBuilder.append("biz_attr="
+ result.biz_attr == null ? "" : result.biz_attr );
        stringBuilder.append("sha=" + result.sha);
        Log.w("TEST",stringBuilder.toString());
    }

    @Override
    public void onFailed(COSRequest cosRequest, final COSResult cosResult) {
        Log.w("TEST", cosResult.code+" : "+cosResult.msg);
     }

});

GetObjectMetadataRequest result=cos.getObjectMetadata(getObjectMetadataRequest);
```

文件删除

方法原型

调用此接口可删除指定文件，具体步骤如下：

1. 调用 DeleteObjectRequest() 方法实例化 DeleteObjectRequest 对象；
2. 调用 COSClient 的 deleteObject 方法，传入 DeleteObjectRequest，回 DeleteObjectResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
appid	String	是	腾讯云APP ID
bucket	String	是	目录所属bucket 名称

参数名称	类型	是否必填	参数描述
cosPath	String	是	远程相对路径
sign	String	是	签名信息，此处单次签名
listener	ICmdTaskListener	否	结果回调

返回结果说明

通过DeleteObjectResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息

示例

```
DeleteObjectRequest deleteObjectRequest = new DeleteObjectRequest();
deleteObjectRequest.setBucket(bucket);
deleteObjectRequest.setCosPath(cosPath);
deleteObjectRequest.setSign(onceSign);
deleteObjectRequest.setListener(new ICmdTaskListener() {
    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
         Log.w("TEST", cosResult.code+" : "+cosResult.msg);
    }

    @Override
    public void onFailed(COSRequest COSRequest, COSResult cosResult) {
        Log.w("TEST", cosResult.code+" : "+cosResult.msg);
   }
});

DeleteObjectResult result = cos.deleteObject(deleteObjectRequest);
```

文件下载

方法原型

调用此接口可下载指定文件，具体步骤如下：

1. 调用 GetObjectRequest(String downloadURl,String savePath) 方法实例化 GetObjectRequest对象；
2. 调用 COSClient 的 getObject 方法，传入 GetObjectRequest,返回 GetObjectResult 对象；

参数说明

参数名称	类型	是否必填	参数描述
downloadUrl	String	是	下载文件的URL
localPath	String	是	本地保存文件的绝对地址
sign	String	否	签名信息：开启了防盗链则需要，否则不需要
listener	IDownloadTaskListener	否	结果回调

返回结果说明

通过GetObjectResult对象的成员变量返回请求结果。

成员变量名称	类型	变量说明
code	String	结果码
msg	String	详细结果信息

示例

```
GetObjectRequest getObjectRequest = new GetObjectRequest(downloadURl,savePath);
getObjectRequest.setSign(null);
getObjectRequest.setListener(new IDownloadTaskListener() {
    @Override
    public void onProgress(COSRequest cosRequest, final long
 currentSize, final long totalSize) {
        float progress =  currentSize / (float)totalSize;
```

```
        progress = progress * 100;
        progressText.setText("progress =" + (int) (progress) + "%");
        Log.w("TEST", "progress =" + (int) (progress) + "%");

    }

    @Override
    public void onSuccess(COSRequest cosRequest, COSResult cosResult) {
         Log.w("TEST","code =" + cosResult.code + "; msg =" + cosResult.msg);
    }

    @Override
    public void onFailed(COSRequest COSRequest, COSResult cosResult) {
                        Log.w("TEST","code ="
 + cosResult.code + "; msg =" + cosResult.msg);
    }
});


GetObjectResul result = cos.getObject(getObjectRequest);
```

C sharp SDK

开发准备

相关资源

github项目

开发准备

1. sdk依赖C# 4.0版本及以上， 推荐使用相同的版本。
2. 从控制台获取APP ID、SecretID、SecretKey。
3. 修改园区，CosCloud.cs文件内的URL定义，
   例如上海：http://sh.file.myqcloud.com/files/v2/，天津：http://tj.file.myqcloud.com/files/v2/。

SDK 配置

直接下载github上提供的源代码，集成到开发环境。

若需 HTTPS 支持，则将 cos_dotnet_sdk/Api/CosCloud.cs 文件中变量 COSAPI_CGI_URL 中的 http 修改为 https 即可。

生成签名

多次有效签名

方法原型

```
public static string Signature(int appId, string secretId, string
 secretKey, long expired, string bucketName)
```

参数说明

参数名	类型	必填	参数描述
appId	int	是	AppId
secretId	string	是	Secret Id
secretKey	string	是	Secret Key，以上三项从控制台获取。
expired	long	是	过期时间，Unix时间戳
bucketName	string	是	bucket名称，bucket创建参见创建Bucket

示例

```
var sign = Sign.Signature(appId, secretId, secretKey, expired, bucketName);
```

单次有效签名

方法原型

```
public static string SignatureOnce(int appId, string secretId, string
 secretKey, string remotePath, string bucketName)
```

参数说明

参数名	类型	必填	参数描述
appId	int	是	AppId
secretId	string	是	Secret Id

参数名	类型	必填	参数描述
secretKey	string	是	Secret Key，以上三项从控制台获取。
bucketName	string	是	bucket名称，bucket创建参见创建Bucket
remotePath	string	是	文件唯一的标识，格式/appid/bucketname/filepath/filename，其中/filepath/filename为文件在此bucketname下的全路径，

示例

```
var
 sign = Sign.SignatureOnce(appId, secretId, secretKey,remotePath, bucketName);
```

更多签名详细说明，请参考权限控制 。

目录操作

创建目录

接口说明：用于目录的创建，可以通过此接口在指定bucket下创建目录。

方法原型

```
public string CreateFolder(string bucketName, string
 remotePath, Dictionary<string, string> parameterDic = null)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	需要创建目录的全路径
parameterDic	string	否	属性字典

属性字典说明

参数名	类型	必填	参数描述
biz_attr	string	否	目录属性

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	否	返回数据，请参考《Restful API 创建目录》

示例

```
var createFolderParasDic = new Dictionary<string, string>();
    createFolderParasDic.Add(CosParameters.PARA_BIZ_ATTR,"new attribute");
var result = cos.CreateFolder(bucketName, folder, createFolderParasDic);
```

目录更新

接口说明：用于目录业务自定义属性的更新，可以通过此接口更新业务的自定义属性字段。

方法原型

```
public string UpdateFolder(string bucketName, string
 remotePath, Dictionary<string, string>  parameterDic = null)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	需要创建目录的全路径
parameterDic	string	是	目录更新参数字典

目录更新参数字典

参数名	类型	必填	参数描述
biz_attr	string	否	目录属性

返回结果说明

参数名	类型	必选	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
var updateParasDic = new Dictionary<string, string>();
    updateParasDic.Add(CosParameters.PARA_BIZ_ATTR,"new attribute");
var result = cos.UpdateFolder(bucketName, folder, updateParasDic);
```

目录查询

接口说明：用于目录属性的查询，可以通过此接口查询目录的属性。

方法原型

```
public string GetFolderStat(string bucketName, string remotePath)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	目录的全路径

返回结果说明

参数名	类型	必选	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	否	目录属性数据，请参考《Restful API 目录查询》

示例

```
var result = cos.GetFolderStat(bucketName, folder);
```

目录删除

接口说明：用于目录的删除，可以通过此接口删除空目录，如果目录中存在有效文件或目录，将不能删除。

方法原型

```
public string DeleteFolder(string bucketName, string remotePath)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	目录的全路径

返回结果说明

参数名	类型	必选	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
var result = cos.DeleteFolder(bucketName, folder);
```

列举目录下文件&目录

接口说明：用于列举目录下文件和目录，可以通过此接口查询目录下的文件和目录属性。

方法原型

```
public string GetFolderList(string bucketName, string
 remotePath, Dictionary<string, string>  parameterDic = null)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	目录的全路径
parameterDic	Dictionary	是	目录列表参数字典

目录列表参数字典

参数名	类型	必填	参数描述
num	string	是	要查询的目录/文件数量，最大1000
listFlag	string	否	listFlag:大于0返回全部数据，否则返回部分数据
context	String	否	透传字段,用于翻页,前端不需理解,需要往后翻页则透传回来
prefix	string	否	搜索前缀

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	API 错误码，成功时为0
message	String	是	错误信息
data	Array	是	返回数据，请参考《Restful API 目录列表》

示例

```
var folderlistParasDic = new Dictionary<string, string>();
    folderlistParasDic.Add(CosParameters.PARA_NUM,"100");
var result = cos.GetFolderList(bucketName, folder, folderlistParasDic);
```

文件操作

文件上传

接口说明：文件上传的统一接口，对于大于8m的文件，内部会通过多次分片的方式进行文件上传。没有断点续传功能，需要自己用分片上传接口实现。

方法原型

```
public string UploadFile(string bucketName, string remotePath, string
 localPath,     Dictionary<string, string>  parameterDic = null)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称，bucket创建参见创建Bucket
remotePath	string	是	文件在服务端的全路径
localPath	string	是	文件本地路径
parameterDic	Dictionary	否	文件上传参数字典

文件上传参数字典

参数名	类型	必填	参数描述
biz_attr	string	否	文件属性
insertOnly	string	否	同名文件是否覆盖；0：覆盖，1：不覆盖（默认）
slice_size	string	否	可选取值为:641024 5121024，11024 1024（ 默认），21024 1024，310241024

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	返回数据，请参考《Restful API 创建文件》

示例

```
var uploadParasDic = new Dictionary<string, string>();
uploadParasDic.Add(CosParameters.PARA_BIZ_ATTR,"file attribute");
uploadParasDic.Add(CosParameters.PARA_INSERT_ONLY,"0");
uploadParasDic.Add(CosParameters.PARA_SLICE_SIZE,SLICE_SIZE.SLIZE_SIZE_3M.ToStri
ng());
result = cos.UploadFile(bucketName, remotePath, localPath, uploadParasDic);
```

文件属性更新

接口说明：用于目录业务自定义属性的更新，可以通过此接口更新业务的自定义属性字段。

方法原型

```
public string UpdateFile(string bucketName, string
 remotePath, Dictionary<string, string> parameterDic = null)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在Cos的全路径
parameterDic	string	否	文件更新参数字典

文件更新参数字典

参数名	类型	必填	参数描述
biz_attr	string	否	待更新的文件属性信息
authority	string	否	eInvalid(继承Bucket的读写权限)；eWRPrivate(私有读写)；eWPrivateRPublic(公有读私有写)
Cache-Control	string	否	指定请求和响应遵循的缓存机制，如：no-cache；max-age=200
Content-Type	string	否	返回内容的MIME类型，如：text/html
Content-Disposition	string	否	控制用户请求所得的内容存为一个文件的时候提供一个默认的文件名，如：attachment; filename="fname.ext"
Content-Language	string	否	使用的语言，如：zh-CN
x-cos-meta-自定义内容	string	否	表示以“x-cos-meta-”名字开头的参数，用户按照自

参数名	类型	必填	参数描述
			身业务场景，设置需要在 Header 中传输什么参数

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
var optionParasDic = new Dictionary<string, string>();
    optionParasDic.Add(CosParameters.PARA_BIZ_ATTR,"new attribute");
    optionParasDic.Add(CosParameters.PARA_AUTHORITY,AUTHORITY.AUTHORITY_PRIVATEP
UBLIC);
    optionParasDic.Add(CosParameters.PARA_CACHE_CONTROL,"no");
    optionParasDic.Add(CosParameters.PARA_CONTENT_TYPE,"application/text");
    optionParasDic
.Add(CosParameters.PARA_CONTENT_DISPOSITION,"filename=\"test.pdf\"");
    optionParasDic.Add(CosParameters.PARA_CONTENT_LANGUAGE,"en");
    optionParasDic.Add("x-cos-meta-test","test");
result = cos.UpdateFile(bucketName, remotePath, optionParasDic);
```

文件查询

接口说明：用于文件的查询，可以通过此接口查询文件的各项属性信息。

方法原型

```
public string GetFileStat(string bucketName, string remotePath)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在Cos的全路径

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	文件属性数据，请参考《Restful API 文件查询》

示例

```
var result = cos.GetFileStat(bucketName, remotePath);
```

文件删除

接口说明：用于文件的删除，可以通过此接口删除已经上传的文件。

方法原型

```
public string DeleteFile(string bucketName, string remotePath)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在服务端的全路径

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
var result = cos.DeleteFile(bucketName, remotePath);
```

分片上传list

接口说明：查询分片上传的情况

方法原型

```
public string UploadSliceList(string bucketName, string remotePath)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在服务端的全路径

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	分片上传信息

上传完成data的数据说明

参数名	类型	必带	参数描述
url	String	是	操作文件的url
access_url	String	是	生成的文件下载url
source_url	String	是	源地址
resource_path	String	是	资源路径. 格式:/appid/bucket/xxx
preview_url	String	否	预览地址

上传未完成data的数据说明

参数名	类型	必带	参数描述
session	String	是	init返回的标识
filesize	Int	是	文件大小
slice_size	Int	是	分片大小（64K-3M） 大于1M 必须为1M 整数倍
sha	String	否	文件的全文sha值，init时带了则返回
listparts	Json Array	是	已上传完成的分片，形如：[{ “offset” :0, “datalen” :1024}, {}, {}].

示例

```
var fileSha = SHA1.GetFileSHA1(localPath);
var result = SliceUploadIni
t(bucketName, remotePath, local
```

```
Path, fileSha, bizAttribute, sliceSize, insertOnly);
```

分片上传init

接口说明：分片上传的第一次握手,如果上一次分片上传未完成，会返回{"code":-4019,"message":"_ERROR_FILE_NOT_FINISH_UPLOAD"}init

接口说明：分片上传的第一次握手

方法原型

```
public string SliceUploadInit(string bucketName, string
 remotePath, string localPath, string fileSha,string
 bizAttribute = "", int
 sliceSize = SLICE_SIZE.SLIZE_SIZE_1M, int insertOnly = 1)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在服务端的全路径
localPath	string	是	本地文件路径
fileSha	stirng	是	文件的Sha值

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0

参数名	类型	必带	参数描述
message	String	是	错误信息
data	Array	是	分片上传信息

data的数据说明

参数名	类型	必带	参数描述
session	string	否	(非秒传的大部分情况会有)唯一标识此文件传输过程的id
slice_size	Int	否	(非秒传大部分情况下会有)分片大小
serial_upload	int	否	(非秒传大部分情况下会有)1：只支持串行分片上传其它：支持并行分片

示例

```
var fileSha = SHA1.GetFileSHA1(localPath);
var result = SliceUploadIni
t(bucketName, remotePath, local
Path, fileSha, bizAttribute, sliceSize, insertOnly);
```

分片上传

接口说明：分片上传数据

方法原型

```
public string SliceUploadData(string bucketName, string
 remotePath, string localPath, string fileSha, string session, long
```

```
offset,int sliceSise,string sign)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在服务端的全路径
localPath	string	是	本地文件路径
fileSha	stirng	是	文件的Sha值
session	string	是	唯一标识此文件传输过程的id
offset	int	是	分片的偏移量
sliceSize	int	是	分片的大小
sign	string	是	签名（多次）

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	分片上传信息

data的数据说明

参数名	类型	必带	参数描述
session	string	是否	(非秒传的大部分情况会有)唯一标识此文件传输过程的id
offset	Int	是	当前分片的offset
datalen	int	是	分片文件长度slice_size
serial_upload	int	否	(非秒传大部分情况下会有)1：只支持串行分片上传其

参数名	类型	必带	参数描述
			它：支持并行分片

示例

```
var fileSha = SHA1.GetFileSHA1(localPath);
var result = SliceUploa
dDataInit(bucketName, remotePath, local
Path, fileSha, session, offset, sliceSize,sign);
```

分片上传 finish

接口说明：告诉COS所有分片上传成功

方法原型

```
public string SliceUploadFinish(string bucketName, string
 remotePath, string localPath, string fileSha, string session)
```

参数说明

参数名	类型	必填	参数描述
bucketName	string	是	bucket名称
remotePath	string	是	文件在服务端的全路径
localPath	string	是	本地文件路径
fileSha	stirng	是	文件的Sha值
session	string	是	唯一标识此文件传输过程的id

返回结果说明

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	分片上传信息

data的数据说明

参数名	类型	必带	参数描述
session	string	是	(非秒传的大部分情况会有)唯一标识此文件传输过程的id
offset	Int	是	当前分片的offset
datalen	int	是	分片文件长度

示例

```
result
 = SliceUploadFinish(bucketName,remotePath,localPath,fileSha, sessionbizAttribut
e, sliceSize, insertOnly);
```

C++ SDK

开发准备

相关资源

github项目

开发环境

1. 安装openssl的库和头文件 http://www.openssl.org/source/
2. 安装curl的库和头文件 http://curl.haxx.se/download/curl-7.43.0.tar.gz
3. 安装jsoncpp的库和头文件 https://github.com/open-source-parsers/jsoncpp
4. 安装boost的库和头文件 http://www.boost.org/
5. 安装cmake工具 http://www.cmake.org/download/
6. 从控制台获取APP ID、SecretID、SecretKey。

注意：

1. sdk中提供了curl和jsoncpp的库以及头文件，以上库编译好后替换掉sdk中相应的库和头文件即可，如果以上库已经安装到系统里，也可删除sdk中相应的库和头文件。
2. curl默认不支持多线程环境，如果项目使用多线程，在编译curl执行 configure 时需指定 --enable-ares 参数来开启异步DNS解析，依赖 c-ares库，如果系统没有，可到http://c-ares.haxx.se/ 下载安装。
3. jsoncpp的1.y.x版本需要c++11的支持，如果编译器不支持，可以换成 0.y.x版本。

SDK 配置

直接下载github上提供的源代码，集成到您的开发环境。

执行下面的命令 ：

```
cd ${cos-cpp-sdk}
mkdir -p build
```

```
cd build
cmake ..
make
```

cos_demo.cpp里面有常见API的例子。生成的cos_demo可以直接运行，生成的静态库名称为：libcossdk.a。生成的 libcossdk.a 放到你自己的工程里lib路径下，include 目录拷贝到自己的工程的include路径下。

生成签名

多次有效签名

方法原型

```
static string AppSignMuti(const uint64_t appId,
                          const string &secretId,
                          const string &secretKey,
                          const uint64_t expired_time,
                          const string &bucketName);
```

参数说明

参数名	类型	是否必填	默认值	参数描述
appId	uint64_t	是	无	项目APP ID
secretId	String	是	无	用户 Secret ID
secretKey	String	是	无	用户 SecretKey
expired_time	uint64_t	否	无	过期时间，Unix时间戳
bucketName	String	否	无	bucket名称

返回结果说明

返回值：签名字符串

示例

```
uint64_t expired = time(NULL) + 60;
string sign = Auth::AppSignMuti(10000000,"SecretId","SecretKey"
,expired,"bucketName");
```

单次有效签名

方法原型

```
static string AppSignOnce(const uint64_t appId,
                          const string &secretId,
                          const string &secretKey,
                          const string &path,
                          const string &bucketName);
```

参数说明

参数名	类型	必须	默认值	参数描述
appId	uint64_t	是	无	项目 APP ID
secretId	String	是	无	项目 SecretID
secretKey	String	是	无	项目 SecretKey
bucketName	String	否	无	bucket名称
path	String	是	无	文件路径，以斜杠开

参数名	类型	必须	默认值	参数描述
				头，例如/filepath/filename，为文件在此bucketname下的全路径

返回值说明

返回值：签名字符串

示例

```
string path= "/myFloder/myFile.rar";
sign = Auth::AppSignOnce(10000000, "SecretId", "SecretKey", path, bucketName);
```

初始化操作

初始化

接口说明：在使用COS操作之前，需要首先进行COS系统参数的设置，然后分别创建CosConfig以及CosAPI对象，COS的操作都是基于CosAPI对象进行的。

配置文件

```
"AppID":********,
"SecretID":"**********************************",
"SecretKey":"**********************************",
"Region":"sh",    //COS区域， 上传和下载域名均是跟此有关系，因此一定要保证正确
"SignExpiredTime":360, //签名超时时间
"CurlConnectTimeoutInms":180,  //http超时时间
```

```
"CurlGlobalConnectTimeoutInms":360, //
"UploadSliceSize":1048576, //文件分片大小，可选值有524288、1048576、2097152、3145728
"IsUploadTakeSha":0, //文件上传时是否携带文件sha值，0：不携带，1：携带
"DownloadDomainType":2, //下载域名类型,1:cdn,2:cos,3:innercos,4:自定义域名
"SelfDomain":"",//自定义域名
"UploadThreadPoolSize":5, //单文件分片上传线程池大小
"AsynThreadPoolSize":2, //异步上传下载线程池大小
"LogoutType":1 //日志输出类型,0:不输出,1:输出到屏幕,2输出到syslog
```

COS API对象构造原型

```
CosConfig(const string& config_file);
CosAPI(CosConfig& config);
```

目录操作

创建目录

接口说明：用于目录的创建，调用者可以通过此接口在指定bucket下创建目录。

方法原型

```
string CosAPI::FolderCreate(FolderCreateReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderCreateReq	无	目录创建请求类型

request成员	类型	默认值	设置方法	描述
bucket	string	无	构造函数	bucket名称
cosPath	string	无	构造函数	上传的目标路径
bizAttr	string	空字符串	构造函数	文件夹属性

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	参数描述
code	Int	返回码，成功时为0
message	String	描述信息
data	Array	返回数据，请参考《Restful API 创建目录》

示例

```
FolderCreateReq folderCreateReq(bucket,folder,folder_biz_attr);
string rsp = cos.FolderCreate(folderCreateReq);
```

目录更新

接口说明：用于目录业务自定义属性的更新，可以通过此接口更新业务的自定义属性字段。

方法原型

```
string FolderUpdate(FolderUpdateReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderUpdateReq	无	目录更新请求类型

参数名	类型	默认值	设置方法	参数描述
bucket	String	无	构造函数	bucket名称
cosPath	String	无	构造函数	目录的全路径
bizAttr	string	空	构造函数	文件夹属性

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	参数描述
code	Int	返回码，成功时为0
message	String	描述信息
data	Array	返回数据，请参考《Restful API 目录更新》

示例

```
FolderUpdateReq folderUpdateReq(bucket,folder, folder_biz_attr);
string rsp = cos.FolderUpdate(folderUpdateReq);
```

目录查询

接口说明：用于目录属性的查询，可以通过此接口查询目录的属性。

方法原型

```
string CosAPI::FolderStat(FolderStatReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderStatReq	无	目录查询请求类型

参数名	类型	默认值	设置方法	参数描述
bucket	String	无	构造函数	bucket名称
cosPath	String	无	构造函数	目录的全路径

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	参数描述
code	Int	返回码，成功时为0
message	String	描述信息

示例

```
FolderStatReq folderStatReq(bucket,folder);
string rsp = cos.FolderStat(folderStatReq);
```

目录删除

接口说明：用于目录的删除，可以通过此接口删除空目录，如果目录中存在有效文件或目录，将不能删除。

方法原型

```
string CosAPI::FolderDelete(FolderDeleteReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderDeleteReq	无	目录删除请求类型

参数名	类型	默认值	设置方法	参数描述
bucket	String	无	构造函数	bucket名称
cosPath	String	无	构造函数	目录的全路径

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	参数描述
code	Int	返回码，成功时为0
message	String	描述信息

示例

```
FolderDeleteReq folderDeleteReq(bucket,folder);
string rsp = cos.FolderDelete(folderDeleteReq);
```

目录列表

接口说明：用于列举目录下文件和目录，可以通过此接口查询目录下的文件和目录属性。

方法原型

```
string CosAPI::FolderList(FolderListReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderListReq	无	目录列表请求类型

参数名	类型	默认值	设置方法	参数描述
bucket	String	无	构造函数	bucket名称
cosPath	String	无	构造函数	目录的全路径
listNum	int	1000	构造函数	查询数目，最大为1000
context	string	空字符串	构造函数	查询上下文。查看第一页，则传空字符串；若需翻页，需要将前一页查询响应中的context设置到参数中

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	是否必然返回	参数描述
code	Int	是	返回码，成功时为0
message	String	是	描述信息
data	Array	是	返回数据，请参考《Restful API 目录列表》

示例

```
FolderListReq folderListReq(bucket,folder);
string rsp = cos.FolderList(folderListReq);
```

文件操作

文件上传

接口说明：文件上传，包括单文件上传和分片上传（大于8M的文件需要通过将文件内容进行分片上传）。

方法原型

```
string CosAPI::FileUpload(FileUploadReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderUploadReq	无	文件上传请求类型

参数名	类型	默认值	设置方法	参数描述
bucket	String	无	构造函数	bucket名字
srcPath	String	无	构造函数	待上传的本地文件路径
cosPath	String	无	构造函数	文件的全路径
bizAttr	string	无	setBizAttr()	文件属性
insertOnly	int	1	构造函数及setInsertOnly()	同名文件是否覆盖。0：表示覆盖同名文件，1：表示不覆盖，当文件存在返回错误
sliceSize	int	1048576	setSliceSize()	分片大小，可选值512K/1M/2M/3M；默认1M，均需转换为字节数值

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	是否必然返回	参数描述
code	Int	是	返回码，成功时为0
message	String	是	描述信息
data	Array	否	返回数据，请参考《Restful API 文件上传》

示例

```
FileUploadReq fileUploadReq(bucket, srcPath, dstPath);
string rsp = cos.FileUpload(fileUploadReq);
```

文件更新

接口说明：用于目录业务自定义属性的更新，可以通过此接口更新业务的自定义属性字段。

方法原型

```
string CosAPI::FileUpdate(FileUpdateReq& request);
```

参数说明

参数名	类型	默认值	参数描述
request	FileUpdateReq	无	文件更新请求类型

参数名	类型	是否必填	设置方法	参数描述
bucket	String	是	构造函数	bucket名称
cosPath	String	是	构造函数	文件在对象存储服务端的路径
bizAttr	string	否	setBizAttr()	文件属性
authority	string	否	setAuthority()	文件权限，默认是继承bucket的权限合法取值: eInvalid(继承bucket), eWRPrivate(私有读写), eWPrivateRPublic(私有写,公有读)
forbid	int	否	setForbid()	文件封禁标志
custom_header	map	否	setCustomHeader()	用户自定义属性

custom_header说明

参数名	类型	是否必填	参数描述
Cache-Control	string	否	参见HTTP的Cache-Control
Content-Type	string	否	参见HTTP的Content-Type
Content-Disposition	string	否	参见HTTP的Content-Language
Content-Language	string	否	参见HTTP的Content-Disposition
Content-Encoding	string	否	参见HTTP的Content-Encoding
x-cos-meta-	string	否	自定义HTTP 头，参数必须以x-cos-meta-开头，值由用户定义，可设置多个

tips: custom_header是整体覆盖式更新，即更新属性可以选择其中的某几个，如果本次只更新其中的某几个，其他的都会被抹掉。

返回结果说明

通过函数返回值返回请求结果的json字符串：

参数名	类型	是否必然返回	参数描述
code	Int	是	返回码，成功时为0
message	String	是	描述信息

示例

```
FileUpdateReq fileUpdateReq(bucket,dstpath);
fileUpdateReq.setBizAttr(file_biz_attr);
fileUpdateReq.setAuthority(auth);
fileUpdateReq.setForbid(0);
fileUpdateReq.setCustomHeader(custom_header);
string rsp = cos.FileUpdate(fileUpdateReq);
```

文件查询

接口说明：用于文件的查询，可以通过此接口查询文件的各项属性信息。

方法原型

```
string CosAPI::FileStat(FileStatReq& request)
```

参数说明

参数名	类型	默认值	参数描述

参数名	类型	默认值	参数描述
request	FileStatReq	无	目录列表请求类型

FileStatReq

参数名	类型	是否必填	默认值	参数描述
bucket	String	是	无	bucket名称
cosPath	String	是	无	文件在对象存储服务端的全路径

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	必然返回	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	返回数据，请参考《Restful API 文件查询》

示例

```
FileStatReq fileStatReq(bucket,dstpath);
string rsp = cos.FileStat(fileStatReq);
```

文件删除

接口说明：用于文件的删除，可以通过此接口删除已经上传的文件。

方法原型

```
FileDeleteReq fileDeleteReq(bucket,dstpath);
string rsp = cos.FileDelete(fileDeleteReq);
```

参数说明

参数名	类型	默认值	参数描述
request	FolderDeleteReq	无	目录删除请求类型

FolderDeleteReq

参数名	类型	是否必填	默认值	参数描述
bucket	String	是	无	bucket名称
cosPath	String	是	无	文件在对象存储服务端的全路径

返回结果说明

通过函数返回值返回请求结果的json字符串

参数名	类型	是否必然返回	参数描述
code	Int	是	返回码，成功时为0
message	String	是	描述信息

示例

```
FileDeleteReq fileDeleteReq(bucket,dstpath);
string rsp = cos.FileDelete(fileDeleteReq);
```

Java SDK

开发准备

相关资源

cos java sdk v4 github项目

环境依赖

JDK 1.7

安装SDK

- maven安装

pom.xml 添加依赖

```
<dependency>
            <groupId>com.qcloud</groupId>
            <artifactId>cos_api</artifactId>
            <version>4.2</version>
</dependency>
```

- 源码安装

从cos java sdk v4 github下载源码

卸载SDK

删除pom依赖或源码

历史版本

4.2版本是针对COS 4.X系统，接口与3.x的基本一致, 如果需要使用历史版本, 请参见cos java sdk v3 github

生成客户端对象

初始化秘钥信息

```
long appId = 1000000;
String secretId = "xxxxxxxxxxxxxxxxxxxxxxxxxxx";
String secretKey = "xxxxxxxxxxxxxxxxxxxxxxxxxxx";
// 设置要操作的bucket
String bucketName = "xxxxxxxxx";
// 初始化秘钥信息
Credentials cred = new Credentials(appId, secretId, secretKey);
```

初始化客户端配置(如设置园区)

```
// 初始化客户端配置
ClientConfig clientConfig = new ClientConfig();
// 设置bucket所在的区域，比如广州(gz)， 天津(tj)
clientConfig.setRegion("gz");
```

生成客户端

```
// 初始化cosClient
```

```
COSClient cosClient = new COSClient(clientConfig, cred);
```

文件操作

上传文件

方法原型

```
String uploadFile(UploadFileRequest request);
```

参数说明

参数名	类型	默认值	参数描述
request	UploadFileRequest	无	上传文件类型请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如/mytest/demo.txt
localPath	String	无	构造函数或set方法	通过磁盘文件上传的本地绝对路径
contentBufer	byte[]	无	构造函数或set方法	通过内存上传的buffer内容
bizAttr	String	空	构造函数或set方法	文件的备注，主要用于对该文件用途的描

request成员	类型	默认值	设置方法	描述
				述
insertOnly	InsertOnly (枚举)	NO_OVER_WRITE (不覆盖)	set方法	是否直插入不覆盖已存在的文件, NO_OVER_WRITE表示只直插入不覆盖, 当文件存在返回错误 OVER_WRITE 表示允许覆盖
enableSavePoint	boolean	true	set方法	是否开启断点文件，开启断点文件后，会在本地记录一个断点, 如果上传失败，则会跳过已经上传过的片。开启断点文件会牺牲一部分上传速度。
enableShaDigest	boolean	false	set方法	是否计算sha摘要，如果开启sha，并且bucket下有相同内容文件，则会触发秒传。sha计算会耗费一定的CPU和时间，建议大文件不开启。
taskNum	int	16	set方法	文件上传的并发数

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
UploadFileRequest uploadFileRequest = new
 UploadFileRequest(bucketName,         "/sample_file.txt", "local_file_1.txt");
```

```
String uploadFileRet = cosClient.uploadFile(uploadFileRequest);
```

下载文件

方法原型

```
String getFileLocal(GetFileLocalRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	GetFileLocalRequest	无	下载文件请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如/mytest/demo.txt
localPath	String	无	构造函数或set方法	要下载到的本地路径
useCDN	boolean	true	set方法	是否通过CDN进行下载
referer	String	空串	set方法	设置Referer(针对开启了refer防盗链的bucket)
rangeStart	long	0	set方法	要下载的字节起始，参见Http Range
rangeEnd	long	Long.MAX_VALUE	set方法	下载的字节结束,

request成员	类型	默认值	设置方法	描述
				参见Http Range

示例

```
String localPathDown = "src/test/resources/local_file_down.txt";
GetFileLocalRequest getFileLocalRequest =
  new GetFileLocalRequest(bucketName, cosFilePath, localPathDown);
getFileLocalRequest.setUseCDN(false);
getFileLocalRequest.setReferer("*.myweb.cn");
String getFileResult = cosClient.getFileLocal(getFileLocalRequest);
```

移动文件

方法原型

```
String moveFile(MoveFileRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	MoveFileRequest	无	移动文件请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如

request成员	类型	默认值	设置方法	描述
				/mytest/demo.txt
dstCosPath	String	无	构造函数或set方法	移动文件的目标地址，必须从bucket下的根/开始，文件路径不能以/结尾，例如/mytest/demo.txt.move
overWrite	枚举类型	NO_OVER_WRITE(不覆盖)	set方法setOverWrite	在移动的目标文件存在时，选择不覆盖还是覆盖，默认不覆盖

示例

```
String cosFilePath = "/sample_file.txt";
String dstCosFilePath = "/sample_file.txt.bak";
MoveFileRequest moveRequest =
  new MoveFileRequest(bucketName, cosFilePath, dstCosFilePath);
String moveFileResult = cosClient.moveFile(moveRequest);
```

获取文件属性

方法原型

```
String statFile(StatFileRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述

参数名	参数类型	默认值	参数描述
request	StatFileRequest	无	获取文件属性请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如 /mytest/demo.txt

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
StatFileRequest stat
FileRequest = new StatFileRequest(bucketName, "/sample_file.txt");
String statFileRet = cosClient.statFile(statFileRequest);
```

更新文件属性

方法原型

```
String updateFile(UpdateFileRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	UpdateFileRequest	无	更新文件属性请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如 /mytest/demo.txt
bizAttr	String	无	set方法	文件的备注，主要用于对改文件用途的描述
authority	String (枚举)	无	set方法	文件权限，默认是继承bucket的权限合法取值: eInvalid(继承bucket), eWRPrivate(私有读写), eWPrivateRPublic(私有写, 公有读)
cacheControl	String	无	set方法	参见HTTP的Cache-Control
contentType	String	无	set方法	参见HTTP的Content-Type
contentLanguage	String	无	set方法	参见HTTP的Content-Language
contentDisposition	String	无	set方法	参见HTTP的Content-Disposition
x-cos-meta-	String	无	set方法	自定义HTTP 头，参数必须以x-cos-meta-开头，值由用户定义，可设置多个

tips: 更新属性可以选择其中的某几个，对于HTTP头部cache_control，content_type,

content_disposition和x-cos-meta-,

如果本次只更新其中的某几个，其他的都会被抹掉，即这4个属性是整体更新。

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
UpdateFileRequest updateFileRequest = new
 UpdateFileRequest(bucketName, "/sample_file.txt");

updateFileRequest.setBizAttr("□□□□");
updateFileRequest.setAuthority(FileAuthority.WPRIVATE);
updateFileRequest.setCacheControl("no cache");
updateFileRequest.setContentDisposition("cos_sample.txt");
updateFileRequest.setContentLanguage("english");
updateFileRequest.setContentType("application/json");
updateFileRequest.setXCosMeta("x-cos-meta-xxx", "xxx");
updateFileRequest.setXCosMeta("x-cos-meta-yyy", "yyy");

String updateFileRet = cosClient.updateFile(updateFileRequest);
```

删除文件

方法原型

```
String delFile(DelFileRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	DelFileRequest	无	删除文件请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如/mytest/demo.txt

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
DelFileRequest delFileRequest = new
 DelFileRequest(bucketName, "/sample_file_move.txt");
String delFileRet = cosClient.delFile(delFileRequest);
```

目录操作

创建目录

方法原型

```
String createFolder(CreateFolderRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	CreateFolderRequest	无	创建目录请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/
bizAttr	String	空	set方法	目录的备注，主要用于对目录用途的描述

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
CreateFolderRequest createFolderReques
t = new CreateFolderRequest(bucketName, "/sample_folder/");
String createFolderRet = cosClient.createFolder(createFolderRequest);
```

获取目录属性

方法原型

```
String statFolder(StatFolderRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	StatFolderRequest	无	获取目录属性请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
StatFolderRequest stat
FolderRequest = new StatFolderRequest(bucketName, "/sample_folder/");
String statFolderRet = cosClient.statFolder(statFolderRequest);
```

更新目录属性

方法原型

```
String updateFolder(UpdateFolderRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	UpdateFolderRequest	无	更新目录属性请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/
bizAttr	String	空	set方法	目录的备注，主要用于对目录用途的描述

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
UpdateFolderRequest updateFolderReques
t = new UpdateFolderRequest(bucketName, "/sample_folder/");
updateFolderRequest.setBizAttr("□□□□□□□□");
String updateFolderRet = cosClient.updateFolder(updateFolderRequest);
```

获取目录列表

方法原型

```
String listFolder(ListFolderRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	ListFolderRequest	无	获取目录成员请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/
num	int	199	构造函数或set方法	获取列表成员的数量，最大为199
prefix	String	空	构造函数或set方法	搜索成员的前缀, 例如prefix为test表示只搜索以test开头的文件或目录
context	String	空	构造函数或set方法	搜索上下文, 由上一次list的结果返回，作为这一次搜索的起点，用于循环获取一个目录下的所有成员

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含成员列表, 详情请参见返回值模块

示例

```
ListFolderRequest listFolderRequest = new
 ListFolderRequest(bucketName, "/sample_folder/");
String listFolderRet = cosClient.listFolder(listFolderRequest);
```

删除目录

方法原型

```
String delFolder(DelFolderRequest request);
```

参数说明

参数名	参数类型	默认值	参数描述
request	DelFolderRequest	无	删除目录请求

request成员	类型	默认值	设置方法	描述
bucketName	String	无	构造函数或set方法	bucket名称
cosPath	String	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾,

request成员	类型	默认值	设置方法	描述
				例如 /mytest/dir/

返回值

返回值类型	返回值描述
String	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
DelFolderRequest delFolderRequest = new
 DelFolderRequest(bucketName, "/sample_folder/");
String delFolderRet = cosClient.delFolder(delFolderRequest);
```

签名管理

签名模块提供了生成多次签名、单次签名和下载签名的接口，其中多次签名和单次签名在文件和目录操作的api内部使用，用户不用关心，下载签名用于方便用户生成下载私有bucket的文件签名。

多次签名

```
String getPeriodEffectiveSign(String bucketName, String
 cosPath, Credentials cred, long expired)
```

使用场景

上传文件, 重命名文件, 创建目录, 获取文件目录属性, 拉取目录列表

参数说明

参数名	参数类型	默认值	参数描述
bucket	String	无	bucket名称
cos_path	String	无	要签名的cos路径
cred	Credentials	无	用户身份信息, 包括appid, secretId, secretkey
expired	long	无	签名过期时间, UNIX时间戳

返回值

base64编码的字符串

示例

```
Credentials cred = new Credentials(appId, secretId, secretKey);
long expired = System.currentTimeMillis() / 1000 + 600;
String
 signStr = Sign.getPeriodEffe
ctiveSign(bucketName, "/pic/test.jpg", cred, expired);
```

单次签名

```
String getOneEffectiveSign(String bucketName, String cosPath, Credentials cred)
```

使用场景

删除和更新文件目录

参数说明

参数名	参数类型	默认值	参数描述
bucket	unicode	无	bucket名称
cos_path	unicode	无	要签名的cos路径
cred	Credentials	无	用户身份信息, 包括appid, secretId, secretkey

返回值

base64编码的字符串

示例

```
Credentials cred = new Credentials(appId, secretId, secretKey);
String signStr = Sign.getOneEffectiveSign(bucketName, "/pic/test.jpg", cred);
```

下载签名

```
String getDownLoadSign(String bucketName, String
 cosPath, Credentials cred, long expired)
```

使用场景

生成文件的下载签名, 用于下载私有bucket的文件

参数说明

参数名	参数类型	默认值	参数描述
bucket	unicode	无	bucket名称
cos_path	unicode	无	要签名的cos路径
cred	Credentials	无	用户身份信息, 包括appid, secretId, secretkey
expired	long	无	签名过期时间, UNIX时间戳

返回值

base64编码的字符串

示例

```
Credentials cred = new Credentials(appId, secretId, secretKey);
long expired = System.currentTimeMillis() / 1000 + 600;
String
 signStr = Sign.getDownLoadSign(bucketName, "/pic/test.jpg", cred, expired);
```

返回值

code	含义
0	操作成功
-1	输入参数错误, 例如输入的本地文件路径不存在, cos文件路径不符合规范
-2	网络错误, 如404等
-3	连接cos时发生异常，如连接超时

PHP SDK

开发准备

相关资源

[cos php sdk v4 github项目]

开发环境

1. 依赖环境：PHP 5.3.0 版本及以上
2. 从控制台获取APP ID、SecretID、SecretKey，并修改 cos-php-sdk-v4/qcloudcos/conf.php 文件里的相关配置。

SDK 配置

下载 SDK 后，在使用 SDK 时，加载 cos-php-sdk-v4/include.php
并设置全局的超时时间及COS所在的区域即可。

```
require('cos-php-sdk-v4/include.php');
use qcloudcos\Cosapi;

Cosapi::setTimeout(180);

// 设置COS所在的区域，对应关系如下:
//      华南   -> gz
//      华中   -> sh
//      华北   -> tj
Cosapi::setRegion('gz');
```

若需要支持 HTTPS ，修改 conf.php 中的 API_COSAPI_END_POINT 的值为如下：

```
const API_COSAPI_END_POINT = 'https://region.file.myqcloud.com/files/v2/';
```

生成签名

多次有效签名

方法原型

```
public static function createReusableSignature($expiration, $bucket, $filepath);
```

参数说明

参数名	类型	必填	参数描述
expiration	long	是	过期时间，Unix时间戳
bucket	String	是	bucket 名称
filepath	String	否	文件路径

示例

```
$expiration = time() + 60;
$bucket = 'testbucket';
$sign = Auth::appSign($expiration, $bucket);
```

单次有效签名

方法原型

```
public static function createNonreusableSignature($bucket, $filepath);
```

参数说明

参数名	类型	必填	参数描述
bucket	String	是	bucket 名称
filepath	String	是	文件路径，以斜杠开头，例如 /filepath/filename，为文件在此 bucketname 下的全路径

示例

```
$bucket = 'testbucket';
$filepath = "/myFloder/myFile.rar";
$sign = Auth::createNonreusableSignature($bucket, $path);
```

目录操作

创建目录

接口说明：用于目录的创建，可通过此接口在指定 bucket 下创建目录。

方法原型

```
public static function createFolder($bucketName, $path, $bizAttr = null);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	目录全路径
bizAttr	String	否	目录属性信息，业务自行维护

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	否	返回数据，请参考《Restful API 创建目录》

示例

```
$bizAttr = "attr_folder";
$result  = Cosapi::createFolder($bu    cketName, $path,$bizAttr)
```

目录更新

接口说明：用于目录业务自定义属性的更新，调用者可以通过此接口更新业务的自定义属性字段。

方法原型

```
public static function updateFolder($bucketName, $path, $bizAttr = null);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	目录路径
bizAttr	String	否	目录属性信息

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
$bizAttr = "folder new attribute";
$result  = Cosapi::updateFolder($bucketName, $path, $bizAttr)
```

目录查询

接口说明：用于目录属性的查询，调用者可以通过此接口查询目录的属性。

原型方法

```
public static function statFolder($bucketName, $path);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	目录路径

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	否	目录 属性数据 ，请参考《Restful API 目录查询》

示例

```
$result = Cosapi::statFolder($bucketName, $path);
```

删除目录

接口说明：用于目录的删除，调用者可以通过此接口删除空目录，如果目录中存在有效文件或目录，将不能删除。

方法原型

```
public static function delFolder($bucketName, $path);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	目录全路径

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
$result = Cosapi::delFolder($bucketName, $path);
```

列举目录中的文件和目录

接口说明：用于列举目录下文件和目录，可以通过此接口查询目录下的文件和目录属性。

方法原型

```
public static function listFolder
($bucketName, $path, $num = 20, $pattern = 'eListBoth', $order = 0
```

```
, $context = null);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket名称
path	String	是	目录的全路径
num	int	否	要查询的目录/文件数量
context	String	否	透传字段，查看第一页，则传空字符串。若需要翻页，需要将前一页返回值中的context透传到参数中。order用于指定翻页顺序。若order填0，则从当前页正序/往下翻页；若order填1，则从当前页倒序/往上翻页
order	int	否	默认正序(=0), 填1为反序
pattern	String	否	eListBoth：列举文件和目录；eListDirOnly：仅列举目录；eListFileOnly：仅列举文件

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	API 错误码，成功时为0
message	String	是	错误信息
data	Array	是	返回数据，请参考《Restful API 目录列表》

示例

```
$result = Cosapi::listFolder($bucketName, $path, 20, 'eListBoth',0);
```

列举目录下指定前缀文件&目录

接口说明：用于列举目录下指定前缀的文件和目录，可以通过此接口查询目录下的指定前缀的文件和目录信息。

原型方法

```
public static function prefixSearch
($bucketName, $prefix, $num = 20, $pattern = 'eListBoth', $order = 0
, $context = null);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket名称，bucket创建参见创建Bucket
prefix	String	是	列出含此前缀的所有文件(带全路径)
num	int	否	要查询的目录/文件数量
context	String	否	透传字段，查看第一页，则传空字符串。若需要翻页，需要将前一页返回值中的context透传到参数中。order用于指定翻页顺序。若order填0，则从当前页正序/往

参数名	类型	必填	参数描述
			下翻页；若order填1，则从当前页倒序/往上翻页
order	int	否	默认正序(=0), 填1为反序
pattern	String	否	eListBoth：列举文件和目录；eListDirOnly：仅列举目录；eListFileOnly：仅列举文件

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	API 错误信息
data	Array	是	返回数据，请参考《Restful API 目录列表》

示例

```
$prefix= "/myFolder/2015-";
$result = Cosapi::prefixSearch($bucketName, $prefix, 20, 'eListBoth',0);
```

文件操作

文件上传

接口说明：文件上传的统一接口，对于大于20M的文件，内部会通过多次分片的方式进行文件上传。

原型方法

```
public static function upload($bucketName, $srcPath, $dstPath,
                $bizAttr = null, $slicesize = null, $insertOnly = null);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket名称，bucket创建参见创建Bucket
srcPath	String	是	本地要上传文件的全路径
dstPath	String	是	文件在COS服务端的全路径，不包括/appid/bucketname
bizAttr	String	否	文件属性，业务端维护
slicesize	int	否	文件分片大小，当文件大于20M时，SDK内部会通过多次分片的方式进行上传，默认分片大小为1M，支持的最大分片大小为3M
insertOnly	int	否	同名文件是否进行覆盖。0：覆盖；1：不覆盖

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	返回数据，请参考《Restful API 创建文件》

示例

```
$dstPath = "/myFolder/test.mp4";
$bizAttr = "";
$insertOnly = 0;
$sliceSize = 3 * 1024 * 1024;
$result = Cosapi::upload($srcPath,$bucketName,dstPath ,"biz_attr");
```

文件属性更新

接口说明：用于目录业务自定义属性的更新，可以通过此接口更新业务的自定义属性字段。

原型方法

```
public static function update($bucketName, $path,

;
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	文件在文件服务端的全路径，不包括/appid/bucketname
bizAttr	String	否	待更新的文件属性信息
authority	String	否	eInvalid(继承Bucket的读

参数名	类型	必填	参数描述
			写权限)；eWRPrivate(私有读写)；eWPrivateRPublic(公有读私有写)
customer_headers_array	String	否	用户自定义头域。可携带参数名分别为：'Cache-Control'、'Content-Type'、'Content-Disposition'、'Content-Language'、以及以'x-cos-meta-'为前缀的参数名称

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
$bizAttr = "";
$authority = "eWPrivateRPublic";
$customer_headers_array = array(
    'Cache-Control' => "no",
    'Content-Language' => "ch",
);
$result = Cosapi::update($bucketName, $dstPath, $bizAttr,$authority, $customer_headers_array);
```

文件查询

接口说明：用于文件的查询，调用者可以通过此接口查询文件的各项属性信息。

原型方法

```
public static function stat($bucketName, $path);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	文件在文件服务端的全路径

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息
data	Array	是	文件 属性数据 ，请参考《Restful API 文件查询》

示例

```
$result = Cosapi::stat($bucketName, $path);
```

文件删除

接口说明：用于文件的删除，调用者可以通过此接口删除已经上传的文件。

原型方法

```
public static function delFile($bucketName, $path);
```

参数说明

参数名	类型	必填	参数描述
bucketName	String	是	bucket 名称
path	String	是	文件的全路径

返回值说明(json)

参数名	类型	必带	参数描述
code	Int	是	错误码，成功时为0
message	String	是	错误信息

示例

```
$result = Cosapi::delFile($bucketName, $path);
```

iOS SDK

开发准备

SDK 获取

对象存储服务的 iOS SDK 的下载地址：iOS SDK

更多示例可参考Demo：iOS Demo

开发准备

- iOS 7.0+；
- 手机必须要有网络（GPRS、3G或Wifi网络等）；
- 从控制台获取APP ID、SecretID、SecretKey。

SDK 配置

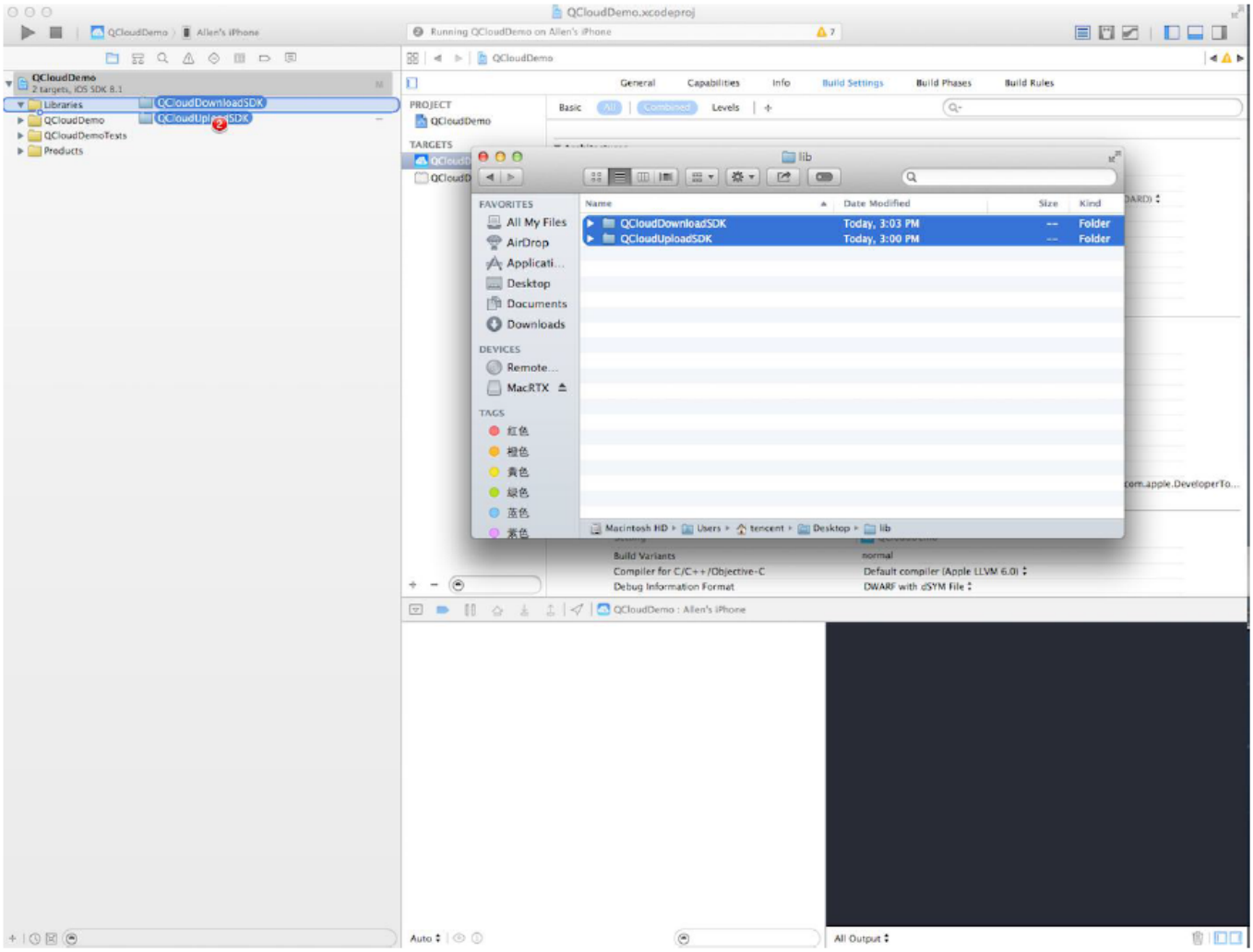
SDK 导入

COS 的 iOS SDK压缩包组成：

- COSClientSDK.zip

压缩包中都包含了一个 .a 静态库和一个包含头文件的文件夹 Headers，如下图所示。上传包提供了支持 bitcode 版本，与不支持 bitcode 版本，可根据业务需要进行选择。

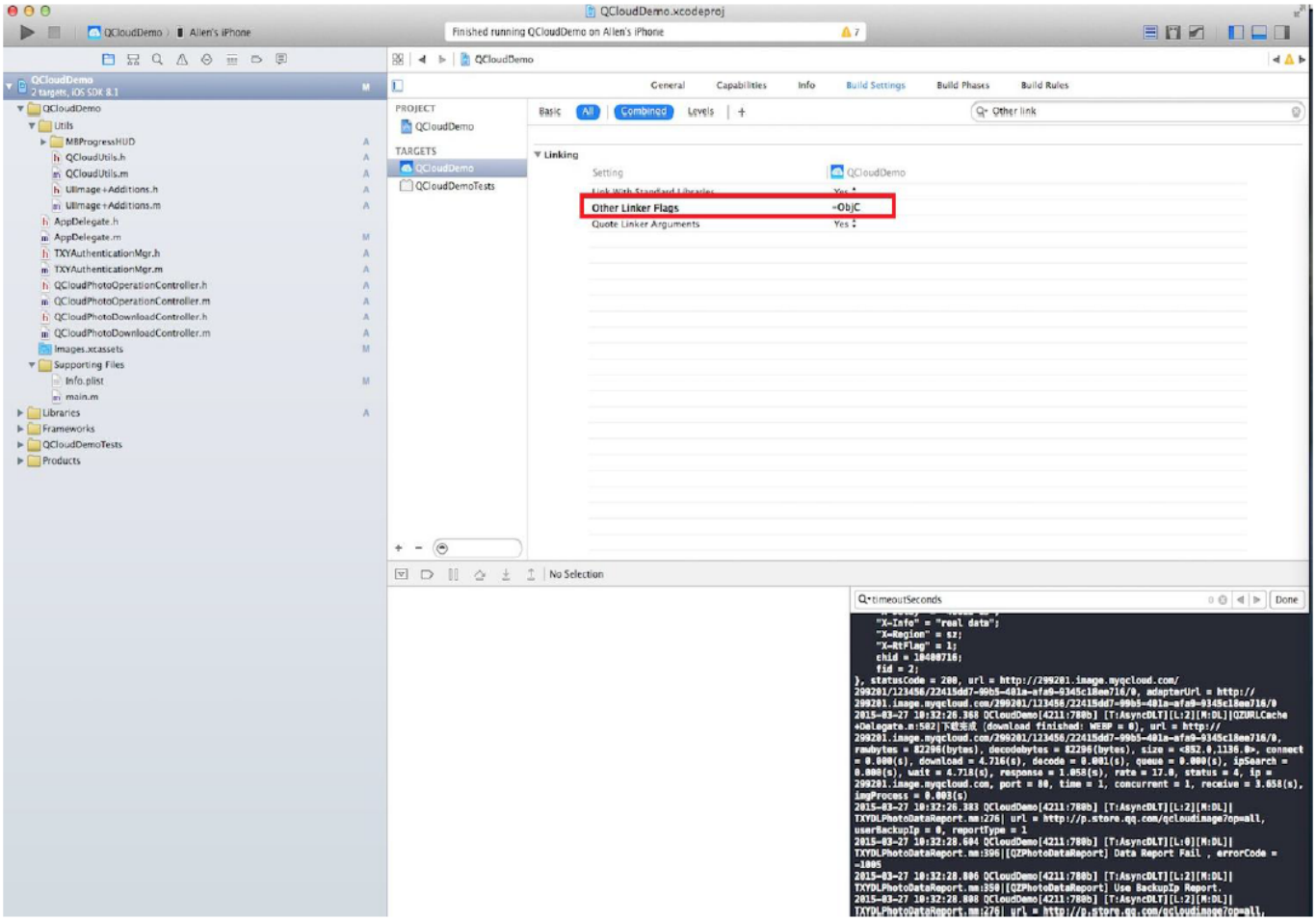
将解压后的 COSSDK 拖入工程目录，Xcode 会自动将其加入链接库列表中。



注意：上传/下载SDK包可根据业务需求选择性导入。

工程配置

在 Build Settings 中设置 Other Linker Flags，加入参数 -ObjC。



在工程info.plist文件中添加App Transport Security Settings 类型，然后在App Transport Security Settings下添加Allow Arbitrary Loads 类型Boolean,值设为YES

初始化

引入上传 SDK 的头文件 COSClient .h，使用目录操作时，需要先实例化 COSClient 对象。

方法原型

```
- (instancetype)initWithAppId:(NSString*)appId  withRegion:(NSString *)region;
```

参数说明

参数名称	类型	是否必填	说明
appId	NSString *	是	项目ID，即APP ID。
region	NSString *	是	bucket被创建的时候机房区域，比如上海： “sh” 广州："gz"

示例

```
COSClient
 *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfig instance].r
egion];
```

快速入门

这里演示的上传和下载的基本流程，更多细节可以参考demo；在进行这一步之前必须在腾讯云控制台上申请COS业务的appid；

STEP - 1 初始化COSClient

示例

```
COSClient
 *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfig instance].r
egion];
```

STEP - 2 上传文件

在这里我们假设您已经申请了自己业务bucket。SDK所有的任务对应了相应的task，只要把相应的task参数传递给client，就可以完成相应的动作；

示例

```
    COSObjectPutTask *task = [COSObjectPutTask new];
    task.filePath = path;
    task.fileName = fileName;
    task.bucket = bucket;
    task.attrs = @"customAttribute";
    task.directory = dir;
    task.insertOnly = YES;
    task.sign = _sign;
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
         //sucess
        }else{}
    };
    client.progressHandler = ^(NSInteger bytesWritten,NSInteger
 totalBytesWritten,NSInteger totalBytesExpectedToWrite){
          //progress
    };
    [client putObject:task];
```

STEP - 3 下载文件

示例

```
     COSObjectGetTask *cm = [[COSObjectGetTask alloc] initWithUrl:imgUrl.text];
```

```
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        //
    };
    client.downloadProgressHandler = ^(int64_t receiveLength,int64_t
 contentLength){
    };
    [client getObjectRequest:cm];
```

生成签名

签名类型：

类型	含义
多次有效	有效时间内多次始终有效
单次有效	与资源URL绑定，一次有效

签名获取：

移动端 SDK 中用到的签名，推荐使用 服务器端SDK，并由移动端向业务服务器请求。

目录操作

目录创建

方法原型

通过此接口在指定的 bucket 下创建目录，具体步骤如下：

1. 实例化 COSCreateDirCommand 对象；
2. 调用 COSClient 的 createDirRequest 方法，将 COSCreateDirCommand 对象传入。
3. 通过COSCreatDirTaskRsp 的对象返回结果

参数说明

参数名称	类型	是否必填	说明
dir	NSString *	是	目录路径（相对于bucket的路径）
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名
attrs	NSString *	否	用户自定义属性

返回结果说明

通过COSCreatDirTaskRsp 的对象返回结果

属性名称	类型	说明
retCode	int	任务描述代码
descMsg	NSString *	任务描述信息

示例

```
    COSCreateDirCommand *cm = [COSCreateDirCommand new];
    cm.directory = dir;
    cm.bucket = bucket;
    cm.sign = _sign;
    cm.attrs = @"dirTest";
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess
        }else{
            //fail
        }
```

```
};
[client createDir:cm];
```

目录属性更新

方法原型

通过调用此接口更新目录的自定义属性，具体步骤如下：

1. 实例化 COSUpdateDirCommand 对象；
2. 调用 COSClient 的 updateDirRequest 方法，将 COSUpdateDirCommand 对象传入；
3. 通过COSUpdateDirTaskRsp 对象返回结果

参数说明

参数名称	类型	是否必填	说明
dir	NSString *	是	目录路径（相对于bucket的路径）
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名
attrs	NSString *	否	用户自定义属性

返回结果说明

通过COSUpdateDirTaskRsp 的对象返回结果

属性名称	类型	说明
retCode	int	任务描述代码
descMsg	NSString *	任务描述信息

示例

```
    COSUpdateDirCommand *cm = [COSUpdateDirCommand new];
    cm.directory = dir;
    cm.bucket = bucket;
    cm.sign = _sign;//□□□□□□□□□□□
    cm.attrs = @"dirTest";
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess
        }else{}
    };
    [client updateDir:cm];
```

目录属性查询

方法原型

通过调用此接口来查询目录的详细属性，具体步骤如下：

1. 实例化 COSDirmMetaCommand 对象；
2. 调用 COSClient 的 getDirMetaData 方法，将 COSDirmMetaCommand 对象传入；
3. 通过COSDirMetaTaskRsp的对象返回结果信息；

参数说明

参数名称	类型	是否必填	说明
dir	NSString *	是	目录路径（相对于bucket的路径）
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名

返回结果说明

通过COSDirmMetaCommand的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码
descMsg	NSString *	任务描述信息
data	NSDictionary *	任务描述信息

示例

```
    COSDirmMetaCommand *cm = [COSDirmMetaCommand new];
    cm.directory = dir;
    cm.bucket = bucket;
    cm.sign = sign;
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess
        }else{}
    };
    [client getDirMetaData:cm];
```

目录删除

方法原型

调用此接口，进行指定 bucket
下目录的删除，如果目录中存在有效文件或目录，将不能删除。具体步骤如下：

1. 实例化 COSDeleteDirCommand 对象；
2. 调用 COSClient 的 deleteDirRequest 命令，传入 COSDeleteDirCommand 对象；
3. 通过COSdeleteDirTaskRsp 的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
dir	NSString *	是	目录路径（相对于bucket的路径）
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名

返回结果说明

通过COSdeleteDirTaskRsp的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码
descMsg	NSString *	任务描述信息

示例

```
    COSDeleteDirCommand *cm = [COSDeleteDirCommand new];
    cm.directory = dir;
    cm.bucket = bucket;
    cm.sign = _oneSign;//□□□□□□□□□□
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess;
        }else{}
    };
    [client deleteDir:cm];
```

目录列表

方法原型

调用此接口可以列出 bucket 中，指定目录下的文件、目录信息，具体步骤如下：

1. 实例化 COSListDirCommand 对象；
2. 调用 COSClient 对象的 listDirRequest 方法，将 COSListDirCommand 对象传入;
3. 通过COSDirListTaskRsp 的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
path	NSString *	是	目录路径（相对于bucket的路径）
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名
num	NSUInteger	是	一次拉取数目设定
pageContext	NSString *	是	透传字段，查看第一页，则传空字符串。若需要翻页，需要将前一页返回值中的context透传到参数中
prefix	NSString *	是	前缀查询

返回结果说明

通过TXYListDirCommandRsp的对象返回结果信息

属性名称	类型	说明
context	NSString *	目录个数
listover	NSString *	文件个数
infos	NSArray *	文件目录属性列表
retCode	int	任务描述代码
descMsg	NSString *	任务描述信息

示例

```
    COSListDirCommand *cm = [COSListDirCommand new]
    cm.directory = dir;
    cm.bucket = bucket;
    cm.sign = _sign;
    cm.number = 100;
    cm.pageContext = @"";
    cm.prefix = @"xx";
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
          //sucess
        }else{}
    };
    [client listDir:cm];
```

文件操作

初始化

方法原型

与目录操作相同，在进行文件操作之前，需引入上传 SDK 的头文件 COSClient .h，实例化 COSClient 对象。

参数说明

参数名称	类型	是否必填	说明
appId	NSString *	是	项目ID，即APP ID。

参数名称	类型	是否必填	说明
region	NSString *	是	bucket被创建的时候机房区域，比如上海：“sh” 广州："gz"

示例

```
- (instancetype)initWithAppId:(NSString*)appId  withRegion:(NSString *)region;
```

文件上传

方法原型

调用此接口者可进行本地文件上传操作，具体步骤如下：

1. 实例化 COSObjectPutTask ；
2. 调用 COSClient 对象的 putObject 方法，将 COSObjectPutTask 对象传入；
3. 通过COSObjectUploadTaskRsp的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
filePath	NSString *	是	文件路径
sign	NSString *	是	签名
bucket	NSString *	是	目标 Bucket 名称
fileName	NSString *	是	目标 文件上传cos后显示的名称
attrs	NSString *	否	文件自定义属性
directory	NSString *	是	文件上传目录，相对路径,举例“/path”
insertOnly	BOOL	是	上传文件的动作是插入覆盖，举例“YES” 文件不会覆

参数名称	类型	是否必填	说明
			盖之前的上传的文件

返回结果说明

通过COSObjectUploadTaskRsp的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息
sourceURL	NSString *	成功后，后台返回文件的 CDN url
sourceURL	NSString *	成功后，后台返回文件的 源站 url

示例

```
    COSObjectPutTask *task = [COSObjectPutTask new];
    task.filePath = path;
    task.fileName = fileName;
    task.bucket = bucket;
    task.attrs = @"customAttribute";
    task.directory = dir;
    task.insertOnly = YES;
    task.sign = _sign;
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
         //sucess
        }else{}
    };
    client.progressHandler = ^(NSInteger bytesWritten,NSInteger
 totalBytesWritten,NSInteger totalBytesExpectedToWrite){
          //progress
    };
```

```
[client putObject:task];
```

文件属性更新

方法原型

调用此接口更新文件的自定义属性，具体步骤如下：

1. 实例化 COSObjectUpdateCommand 对象；
2. 调用 COSClient 的 updateObject 命令，传入 COSObjectUpdateCommand 对象；
3. 通过COSObjectUpdateTaskRsp的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
fileName	NSString *	是	
bucket	NSString *	是	目录所属 bucket 名称
sign	NSString *	是	签名
attrs	NSString *	否	用户自定义属性

返回结果说明

通过TXYUpdateCommandRsp的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息

示例

```
COSObjectUpdateCommand *cm = [COSObjectUpdateCommand new]
```

```
    cm.fileName = file;
    cm.bucket = bucket;
    cm.sign = _oneSign;//□□□□
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess
        }else{
    };
    [client updateObject:cm];
```

文件属性查询

方法原型

调用此接口可查询文件的属性信息，具体步骤如下：

1. 实例化 COSObjectMetaCommand 对象；
2. 调用 COSClient 的 getObjectInfo 命令，传入 COSObjectMetaCommand 对象；
3. 通过COSObjectMetaTaskRsp 类返回结果信息

参数说明

参数名称	类型	是否必填	说明
filename	NSString *	是	
bucket	NSString *	是	文件所属 bucket 名称
directory	NSString *	是	目录路径（相对于bucket的路径）
sign	NSString *	是	签名

返回结果说明

通过TXYStatCommandRsp类返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息
data	NSDictionary *	成功时，文件基本信息

示例

```
    COSObjectMetaCommand *cm = [COSObjectMetaCommand new] ;
    cm.fileName = file;
    cm.bucket = bucket;
    cm.directory = dir;
    cm.sign = _oneSign;//□□□□
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
             //sucess
        }else{}
    };
    [client getObjectMetaData:cm];
```

文件删除

方法原型

调用此接口进行文件的删除操作，具体步骤如下：

1. 实例化 COSObjectDeleteCommand 对象；

2. 调用 COSClient 的 deleteObject 命令，传入 COSObjectDeleteCommand 对象。

3. 通过COSObjectDeleteTaskRsp的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
filename	NSString *	是	
bucket	NSString *	是	文件所属 Bucket 名称
directory	NSString *	是	目录路径（相对于bucket的路径）
sign	NSString *	是	签名
objectType	TXYObjectType	是	业务类型，文件删除时设置为：TXYObjectFile

返回结果说明

通过COSObjectDeleteTaskRsp的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息

示例

```
    COSObjectDeleteCommand *cm = [COSObjectDeleteCommand new]□
    cm.fileName = file;
    cm.bucket = bucket;
    cm.directory = dir;
    cm.sign = _oneSign;//□□□□
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance].region];;
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        if (resp.retCode == 0) {
            //sucess
        }else{        }
```

```
};
[client deleteObject:cm];
```

文件下载

方法原型

调用此接口进行文件的下载操作，具体步骤如下：

1. 实例化 COSObjectGetTask 对象；
2. 调用 COSClient 的 getObjectRequest 命令，传入 COSObjectGetTask 对象。
3. 通过COSGetObjectTaskRsp 的对象返回结果信息

参数说明

参数名称	类型	是否必填	说明
filePath	NSString *	是	文件下载地址

返回结果说明

通过 通过COSGetObjectTaskRsp 的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息
object	NSMutableData *	下载文件

示例

```
COSObjectGetTask *cm = [[COSObjectGetTask alloc] initWithUrl:imgUrl.text];
COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
```

```
g instance].region];
    client.completionHandler = ^(COSTaskRsp *resp, NSDictionary *context){
        //
    };
    client.downloadProgressHandler = ^(int64_t receiveLength,int64_t
 contentLength){
    };
    [client getObject:cm];
```

文件分片上传

方法原型

调用此接口进行文件的下载操作，具体步骤如下：

1. 实例化 COSObjectPutTask 对象；
2. 调用 COSClient 的 putObject 命令，传入 COSObjectPutTask 对象。
3. 通过COSObjectUploadTaskRsp 的对象返回结果信息
4. 当multipartUpload 参数设置为YES 的时候上传文件上传的方式为分片上传，该参数默认为NO；

参数说明

参数名称	类型	是否必填	说明
filePath	NSString *	是	文件路径
multipartUpload	BOOL	否	文件上传是否使用分片上传
sign	NSString *	是	签名
bucket	NSString *	是	目标 Bucket 名称
fileName	NSString *	是	目标 文件上传cos后显示的名称
attrs	NSString *	否	文件自定义属性
directory	NSString *	是	文件上传目录，相对路径

参数名称	类型	是否必填	说明
			,举例“/path”
insertOnly	BOOL	是	上传文件的动作是插入覆盖，举例“YES” 文件不会覆盖之前的上传的文件

返回结果说明

通过COSObjectUploadTaskRsp 的对象返回结果信息

属性名称	类型	说明
retCode	int	任务描述代码，为retCode >= 0时标示成功，为负数表示为失败
descMsg	NSString *	任务描述信息

示例

```
    COSObjectPutTask *task = [[COSObjectPutTask alloc] init];
    task.multipartUpload = YES;//□□□□□□□□□
    task.filePath = path;
    task.fileName = fileName;
    task.bucket = bucket;
    task.attrs = @"customAttribute";
    task.directory = dir;
    task.insertOnly = YES;
    task.sign = _sign;
    COSClient *client= [[COSClient alloc] initWithAppId:appId withRegion:[Congfi
g instance]
.region];  client.comp
letionHandler = ^(COSTaskRsp *resp, NSDictionary *context){

        if (rsp.retCode == 0) {
          //sucess
        }else{ }
    };
```

```
    client.progressHandler = ^(NSInteger bytesWritten,NSInteger
totalBytesWritten,NSInteger totalBytesExpectedToWrite){
      //□□
    };
    [client putObject:task];
```

Python SDK

开发准备

相关资源

- GitHub GitHub项目地址，欢迎贡献代码以及反馈问题。
- PyPi PyPi项目地址

环境依赖

python 2.7

获取python版本的方法:

- Linux Shell

```
$ python -V
Python 2.7.11
```

- Windows cmd

```
D:\>python -V
Python 2.7.11
```

如果提示不是内部或外部命令，请现在windows环境变量PATH里添加上python的绝对路径

安装SDK

- pip安装

```
pip install qcloud_cos_v4
```

- 源码安装

github上下载SDK, 解压后如下执行(如果提示permission deny, 需要有管理员权限)

```
python setup.py install
```

卸载SDK

```
pip uninstall qcloud_cos_v4
```

历史版本

3.3版本对接口等进行了重构，和之前的历史版本诸多不同，同时修复了一些bug, 和历史版本不兼容,
如果需要使用历史版本, 请参见[github文件操作

生成客户端对象

初始化客户端

```
    appid = 100000                  # □□□□□□appid
```

```
secret_id = u'xxxxxxxx'          # 替换为用户的secret_id
secret_key = u'xxxxxxx'          # 替换为用户的secret_key
region = "shanghai" #            # 替换为用户的region，目前可以为 shanghai/guangzhou
cos_client = CosClient(appid, secret_id, secret_key, region)
```

注：region参数用来设置园区，目前可以 shanghai或者guangzhou

自定义接入点

```
conf = CosConfig(hostname='', download_hostname='')
cos_client.set_config(conf)
```

文件操作

上传文件

方法原型

```
def upload_file(self, request)
```

参数说明

参数名	类型	默认值	参数描述

参数名	类型	默认值	参数描述
request	UploadFileRequest	无	上传文件类型请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如 /mytest/demo.txt
local_path	unicode	无	构造函数或set方法	要上传的本地文件的绝对路径
biz_attr	unicode	空	构造函数或set方法	文件的备注，主要用于对该文件用途的描述
insert_only	int (枚举)	1	构造函数或set方法	是否直插入不覆盖已存在的文件, 1表示只直插入不覆盖, 当文件存在返回错误 0 表示允许覆盖

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
    request = UploadFileRequest(bucket, u'/sample_file.txt',
u'local_file_1.txt')
    upload_file_ret = cos_client.upload_file(request)
```

获取文件属性

方法原型

```
def stat_file(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	StatFileRequest	无	获取文件属性请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如/mytest/demo.txt

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
request = StatFileRequest(bucket, u'/sample_file.txt')
stat_file_ret = cos_client.stat_file(request)
```

更新文件属性

方法原型

```
def update_file(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	UpdateFileRequest	无	更新文件属性请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如/mytest/demo.txt
biz_attr	unicode	无	set方法	文件的备注，主要用于对改文件用途的描述
authority	unicode (枚举)	无	set方法	文件权限，默认是继承bucket的权限合法取值: eInvalid(继承bucket), eWRPrivate(私有读写), eWPrivateRPublic(私有写, 公有读)
cache_control	unicode	无	set方法	参见HTTP的Cache-Control
content_type	unicode	无	set方法	参见HTTP的Content-Type

request成员	类型	默认值	设置方法	描述
content_language	unicode	无	set方法	参见HTTP的Content-Language
content_disposition	unicode	无	set方法	参见HTTP的Content-Disposition
x-cos-meta-	unicode	无	set方法	自定义HTTP 头，参数必须以x-cos-meta-开头，值由用户定义，可设置多个

tips: 更新属性可以选择其中的某几个，对于HTTP头部cache_control，content_type, content_disposition和x-cos-meta-,
如果本次只更新其中的某几个，其他的都会被抹掉，即这4个属性是整体更新。

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
request = UpdateFileRequest(bucket, u'/sample_file.txt')

request.set_biz_attr(u'这是个demo文件')              # 设置文件biz_attr属性
request.set_authority(u'eWRPrivate')                 # 设置文件的权限
request.set_cache_control(u'cache_xxx')              # 设置Cache-Control
request.set_content_type(u'application/text')        # 设置Content-Type
request.set_content_disposition(u'ccccxxx.txt')      # 设置Content-Disposition
request.set_content_language(u'english')             # 设置Content-Language
request.set_x_cos_meta(u'x-cos-meta-xxx', u'xxx')    # 设置自定义的x-cos-meta-属性
request.set_x_cos_meta(u'x-cos-meta-yyy', u'yyy')    # 设置自定义的x-cos-meta-属性

update_file_ret = cos_client.update_file(request)
```

删除文件

方法原型

```
def del_file(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	DelFileRequest	无	删除文件请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，文件路径不能以/结尾, 例如 /mytest/demo.txt

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
    request = DelFileRequest(bucket, u'/sample_file_move.txt')
```

```
del_ret = cos_client.del_file(request)
```

目录操作

创建目录

方法原型

```
def create_folder(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	CreateFolderRequest	无	创建目录请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/
biz_attr	unicode	空	set方法	目录的备注，主要用于对目录用途的描述

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因,

返回值类型	返回值描述
	详情请参见返回值模块

示例

```
request = CreateFolderRequest(bucket, u'/sample_folder/')
create_folder_ret = cos_client.create_folder(request)
```

获取目录属性

方法原型

```
def stat_folder(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	StatFolderRequest	无	获取目录属性请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess, 'data':\$data},

返回值类型	返回值描述
	code为0表示成功, message为SUCCESS或者失败原因, data中包含相关的属性, 详情请参见返回值模块

示例

```
request = StatFolderRequest(bucket, u'/sample_folder/')
stat_folder_ret = cos_client.stat_folder(request)
```

更新目录属性

方法原型

```
def update_folder(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	UpdateFolderRequest	无	更新目录属性请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/
biz_attr	unicode	空	set方法	目录的备注，主要用于对目录用途的描述

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
request = UpdateFolderRequest(bucket, u'/sample_folder/')
request.set_biz_attr(u'□□□□□□□□□')
update_folder_ret = cos_client.update_folder(request)
```

获取目录列表

方法原型

```
def list_folder(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	ListFolderRequest	无	获取目录成员请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾,

request成员	类型	默认值	设置方法	描述
				例如 /mytest/dir/
num	int	199	构造函数或set方法	获取列表成员的数量，最大为199
prefix	unicode	空	构造函数或set方法	搜索成员的前缀, 例如prefix为test表示只搜索以test开头的文件或目录
context	unicode	空	构造函数或set方法	搜索上下文, 由上一次list的结果返回，作为这一次搜索的起点，用于循环获取一个目录下的所有成员

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess, 'data':\$data}, code为0表示成功, message为SUCCESS或者失败原因, data中包含成员列表, 详情请参见返回值模块

示例

```
request = ListFolderRequest(bucket, u'/sample_folder/')
list_folder_ret = cos_client.list_folder(request)
```

删除目录

方法原型

```
def del_folder(self, request)
```

参数说明

参数名	参数类型	默认值	参数描述
request	DelFolderRequest	无	删除目录请求

request成员	类型	默认值	设置方法	描述
bucket_name	unicode	无	构造函数或set方法	bucket名称
cos_path	unicode	无	构造函数或set方法	cos路径, 必须从bucket下的根/开始，目录路径必须以/结尾, 例如 /mytest/dir/

返回值

返回值类型	返回值描述
dict	{'code':\$code, 'message':$mess}, code为0表示成功, message为SUCCESS或者失败原因, 详情请参见返回值模块

示例

```
request = DelFolderRequest(bucket, u'/sample_folder/')
delete_folder_ret = cos_client.del_folder(request)
```

签名管理

签名模块提供了生成多次签名、单次签名和下载签名的接口，其中多次签名和单次签名在文件和目录操作的api内部使用，用户不用关心，下载签名用于方便用户生成下载私有bucket的文件签名。

多次签名

```
def sign_more(self, bucket, cos_path, expired)
```

使用场景

上传文件, 重命名文件, 创建目录, 获取文件目录属性, 拉取目录列表

参数说明

参数名	参数类型	默认值	参数描述
bucket	unicode	无	bucket名称
cos_path	unicode	无	要签名的cos路径
expired	int	无	签名过期时间, UNIX时间戳

返回值

base64编码的字符串

示例

```
cred = CredInfo(100000, u'xxxxxxx', u'xxxxxxxx')
# appid, secret_id, secret_key
auth_obj = Auth(cred)
sign_str = auth_obj.sign_more(u'mybucket', u'/pic/1.jpg'
,  int(time.time()) + 600)
```

单次签名

```
def sign_once(self, bucket, cos_path)
```

使用场景

删除和更新文件目录

参数说明

参数名	参数类型	默认值	参数描述
bucket	unicode	无	bucket名称
cos_path	unicode	无	要签名的cos路径

返回值

base64编码的字符串

示例

```
cred = CredInfo(100000, u'xxxxxxx', u'xxxxxxxx')
# appid, secret_id, secret_key
auth_obj = Auth(cred)
sign_str = auth_obj.sign_once(u'mybucket', u'/pic/1.jpg')
```

下载签名

```
def sign_download(self, bucket, cos_path, expired)
```

使用场景

生成文件的下载签名, 用于下载私有bucket的文件

参数说明

参数名	参数类型	默认值	参数描述
bucket	unicode	无	bucket名称
cos_path	unicode	无	要签名的cos路径
expired	int	无	签名过期时间, UNIX时间戳

返回值

base64编码的字符串

示例

```
cred = CredInfo(100000, u'xxxxxxx', u'xxxxxxxx')
# appid, secret_id, secret_key
auth_obj = Auth(cred)
sign_str = auth_obj.sign_download(u'mybucket', u'/pic/1.jpg'
,  int(time.time()) + 600)
```

返回码

code	含义
0	操作成功
-1	输入参数错误, 例如输入的本地文件路径不存在, cos文件路径不符合规范

code	含义
-2	网络错误, 如404等
-3	连接cos时发生异常，如连接超时
-71	操作频率过快，触发cos的攻击