

短视频 视频上传 产品文档



腾讯云

【版权声明】

©2013-2018 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模式的承诺或保证。

文档目录

视频上传

 签名派发

 视频上传 (iOS)

 视频上传 (Android)

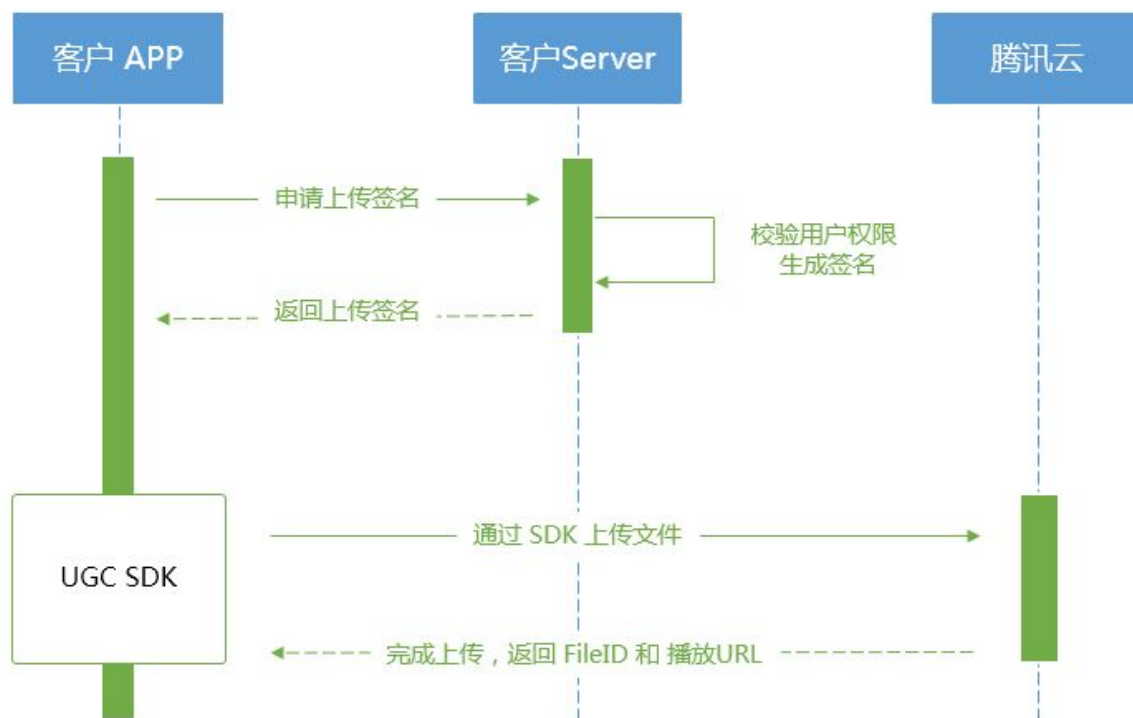
视频上传 签名派发

最近更新时间：2018-05-15 12:23:48

客户端视频上传，是指 App 的最终用户将本地视频直接上传到腾讯云点播。客户端上传的详细介绍请参考点播[客户端上传指引](#)，本文将以最简洁的方式介绍客户端上传的签名生成方法。

总体介绍

客户端上传的整体流程如下图所示：



为了支持客户端上传，开发者需要搭建两个后台服务：签名派发服务和事件通知接收服务。

- 客户端首先向签名派发服务请求上传签名；
- 签名派发服务校验该用户是否有上传权限，若校验通过，则生成签名并下发；否则返回错误码，上传流程结束；
- 客户端拿到签名后使用短视频 SDK 中集成的上传功能来上传视频；
- 上传完成后，点播后台会发送[上传完成事件通知](#)给开发者的事件通知接收服务；

- 如果签名派发服务在签名中指定了视频处理[任务流](#)，点播服务会在视频上传完成后根据指定流程自动进行视频处理。短视频场景下的视频处理一般为 [AI 鉴黄](#)；
- 视频处理完成之后，点播后台会发送[任务流状态变更事件通知](#)给开发者的事件通知接收服务；

至此整个视频上传-处理流程结束。

签名生成

有关客户端上传签名的详细介绍请参考点播[客户端上传签名](#)。

签名派发服务实现示例

```
/**
 * 计算签名
 */
function createFileUploadSignature({ timeStamp = 86400, procedure = '', classId = 0, oneTimeValid = 0, sourceContext = '' }) {
  // 确定签名的当前时间和失效时间
  let current = parseInt((new Date()).getTime() / 1000)
  let expired = current + timeStamp; // 签名有效期：1天
  // 向参数列表填入参数
  let arg_list = {
    //required
    secretId: this.conf.SecretId,
    currentTimeStamp: current,
    expireTime: expired,
    random: Math.round(Math.random() * Math.pow(2, 32)),
    //opts
    procedure,
    classId,
    oneTimeValid,
    sourceContext
  }
  // 计算签名
  let original = querystring.stringify(arg_list);
  let original_buffer = new Buffer(original, "utf8");
  let hmac = crypto.createHmac("sha1", this.conf.SecretKey);
  let hmac_buffer = hmac.update(original_buffer).digest();
  let signature = Buffer.concat([hmac_buffer, original_buffer]).toString("base64");
  return signature;
}
```

```
/**
 * 响应签名请求
 */
function getUploadSignature(req, res) {
  res.json({
    code: 0,
    message: 'ok',
    data: {
      //上传时指明模版参数与任务流
      signature: gVodHelper.createFileUploadSignature({ procedure: 'QCVB_SimpleProcessFile({1,1,1,1})' })
    }
  });
}
```

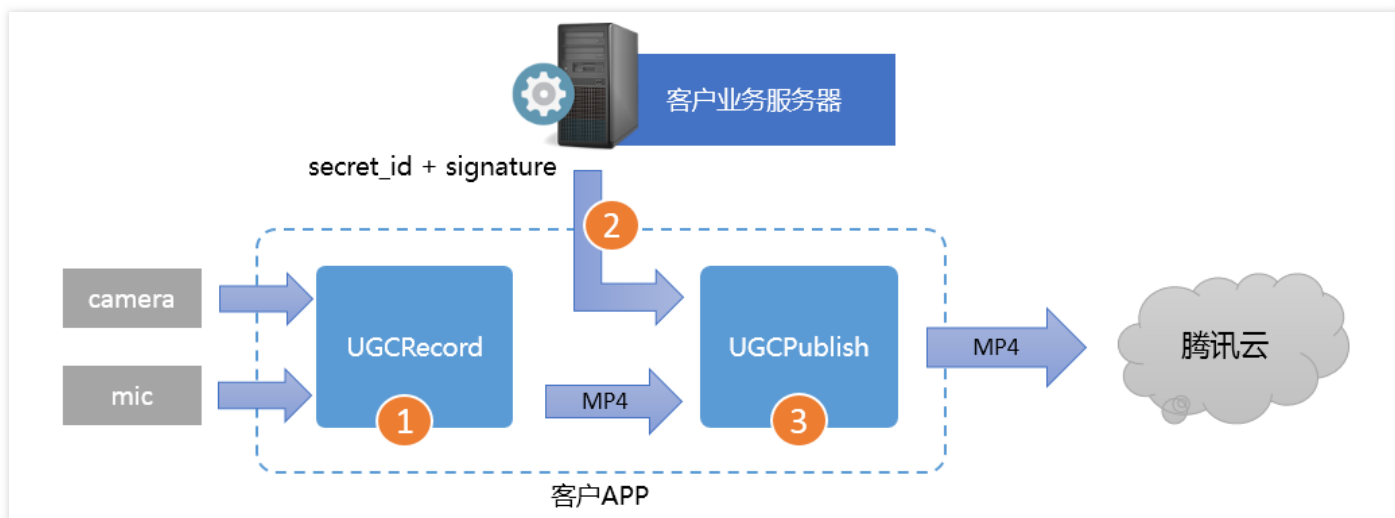
视频上传 (iOS)

最近更新时间：2018-05-15 12:25:24

对接流程

• 短视频发布：

即将 MP4 文件上传到腾讯视频云，并获得在线观看 URL，腾讯视频云支持视频观看的就近调度、秒开播放、动态加速 以及海外接入等要求，从而确保优质的观看体验。

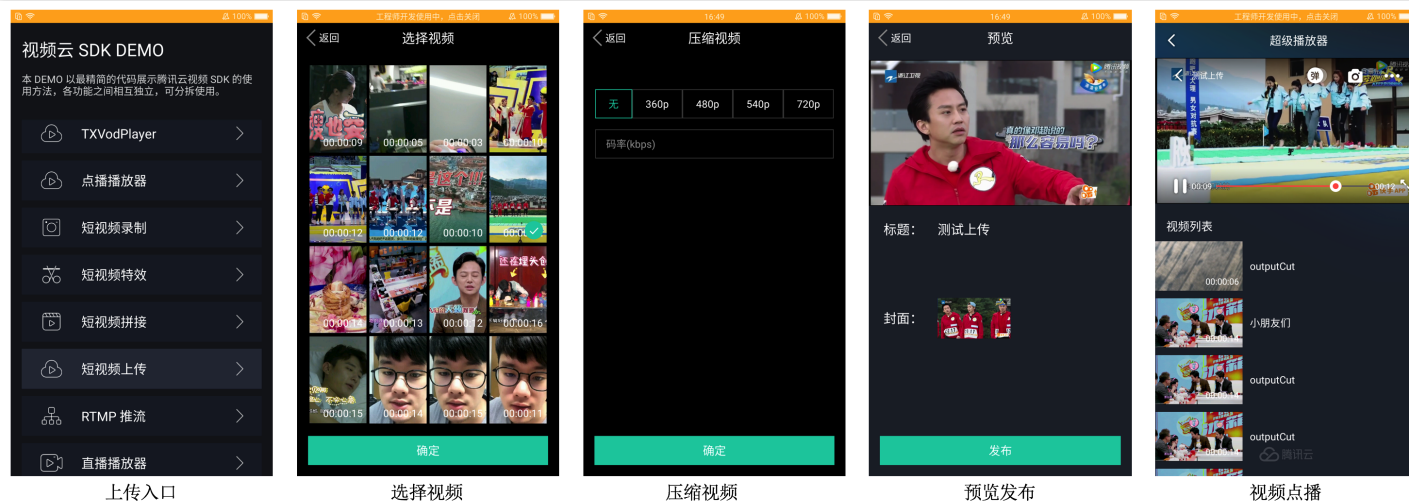


- 第一步：使用 TXUGCRecord 接口录制一段小视频，录制结束后会生成一个小视频文件（MP4）回调给客户；
- 第二步：您的 App 向您的业务服务器申请上传签名。上传签名是 App 将 MP4 文件上传到腾讯云视频分发平台的“许可证”，为了确保安全性，这些上传签名都要求由您的业务 Server 进行签发，而不能由终端 App 生成。
- 第三步：使用 TXUGCPublish 接口发布视频，发布成功后 SDK 会将观看地址的 URL 回调给您。

特别注意

- App 千万不要把计算上传签名的 SecretID 和 SecretKey 写在客户端的代码里，这两个关键信息泄露将导致安全隐患，比如恶意攻击者一旦破解 App 获取该信息，就可以免费使用您的流量和存储服务。
- 正确的做法是在您的服务器上用 SecretID 和 SecretKey 生成一次性的上传签名然后将签名交给 App。因为服务器一般很难被攻陷，所以安全性是可以保证的。
- 发布短视频时，请务必保证正确传递 Signature 字段，否则会发布失败；

对接攻略



1、选择视频

可以接着上篇文档中的录制或者编辑，把生成的视频进行上传，或者可以选择手机本地的视频进行上传。

2、压缩视频

对选择的视频进行压缩，使用 `TXVideoEditor.generateVideo(int videoCompressed, String videoOutputPath)` 接口，支持 4 种分辨率的压缩，后续会增加自定义码率的压缩。

3、发布视频

把刚才生成的 MP4 文件发布到腾讯云上，App 需要拿到上传文件用的短期有效上传签名，这部分有独立的文档介绍，详情请参考 [签名派发](#)。

TXUGCPublish (位于 TXUGCPublish.h) 负责将 MP4 文件发布到腾讯云视频分发平台上，以确保视频观看的就近调度、秒开播放、动态加速 以及海外接入等需求。

```
TXPublishParam * param = [[TXPublishParam alloc] init];
```

```
param.signature = _signature; // 需要填写第四步中计算的上传签名
```

```
// 录制生成的视频文件路径 TXVideoRecordListener 的 onRecordComplete 回调中可以获取
```

```
param.videoPath = _videoPath;
```

```
// 录制生成的视频首帧预览图路径。值为通过调用startRecord指定的封面路径，或者指定一个路径，然后将TXVideoRecordListener 的 onRecordComplete 回调中获取到的UIImage保存到指定路径下，可以置为 nil。
```

```
param.coverPath = _coverPath;
```

```
TXUGCPublish *_ugcPublish = [[TXUGCPublish alloc] init];
```

```
// 文件发布默认是采用断点续传
```

```
_ugcPublish.delegate = self; // 设置 TXVideoPublishListener 回调  
[_ugcPublish publishVideo:param];
```

发布的过程和结果是通过 TXVideoPublishListener (位于 TXUGCPublishListener.h 头文件中定义) 接口反馈出来的：

- onPublishProgress 用于反馈文件发布的进度，参数 uploadBytes 表示已经上传的字节数，参数 totalBytes 表示需要上传的总字节数。

@optional

```
-(void) onPublishProgress:(NSInteger)uploadBytes totalBytes: (NSInteger)totalBytes;
```

- onPublishComplete 用于反馈发布结果，TXPublishResult 的字段 errCode 和 descMsg 分别表示错误码和错误描述信息，videoURL 表示短视频的点播地址，coverURL 表示视频封面的云存储地址，videoId 表示视频文件云存储 Id，您可以通过这个Id调用点播 [服务端API接口](#)。

@optional

```
-(void) onPublishComplete:(TXPublishResult*)result;
```

- 发布结果
通过 [错误码表](#) 来确认短视频发布的结果。

4、播放视频

- 第3步上传成功后，会返回视频的 fileId，播放地址 url，封面 url。用 [点播播放器](#)可以直接传入 fileId 播放，或者 url 播放。

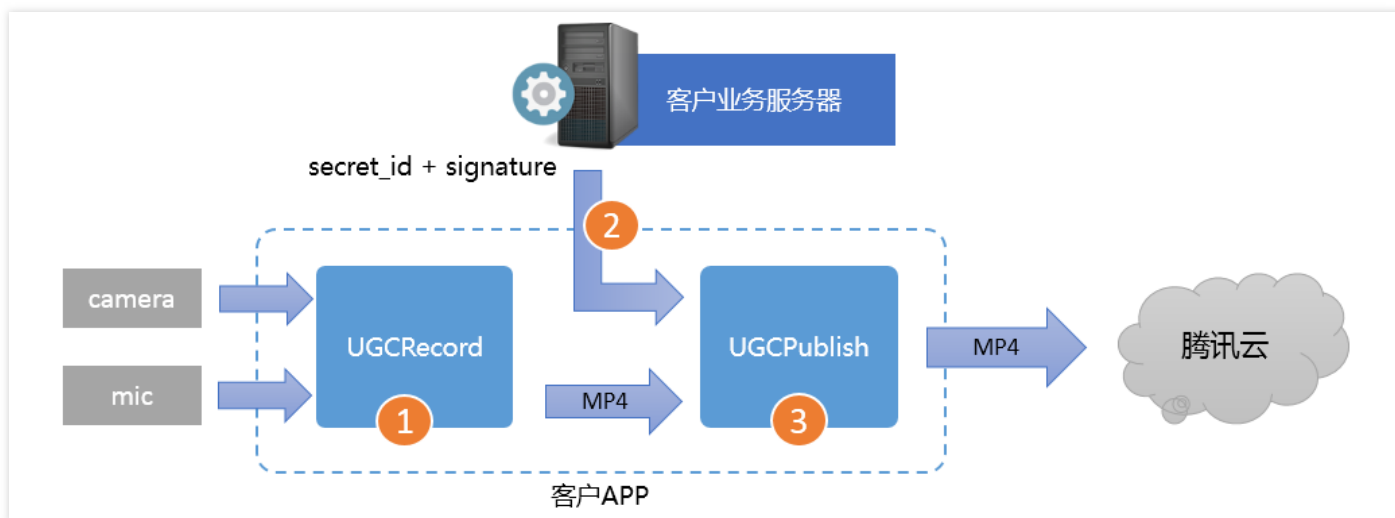
视频上传 (Android)

最近更新时间：2018-05-15 12:24:46

对接流程

• 短视频发布：

即将 MP4 文件上传到腾讯视频云，并获得在线观看 URL，腾讯视频云支持视频观看的就近调度、秒开播放、动态加速 以及海外接入等要求，从而确保优质的观看体验。

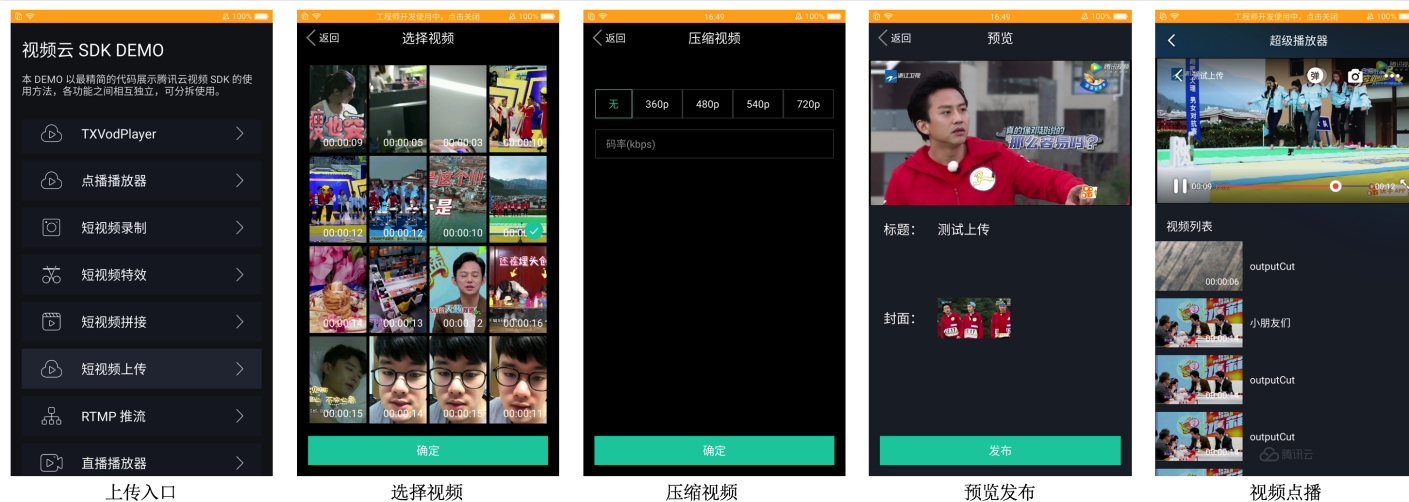


- 第一步：使用 TXUGCRecord 接口录制一段小视频，录制结束后会生成一个小视频文件（MP4）回调给客户；
- 第二步：您的 App 向您的业务服务器申请上传签名。上传签名是 App 将 MP4 文件上传到腾讯云视频分发平台的“许可证”，为了确保安全性，这些上传签名都要求由您的业务 Server 进行签发，而不能由终端 App 生成。
- 第三步：使用 TXUGCPublish 接口发布视频，发布成功后 SDK 会将观看地址的 URL 回调给您。

特别注意

- App 千万不要把计算上传签名的 SecretID 和 SecretKey 写在客户端的代码里，这两个关键信息泄露将导致安全隐患，比如恶意攻击者一旦破解 App 获取该信息，就可以免费使用您的流量和存储服务。
- 正确的做法是在您的服务器上用 SecretID 和 SecretKey 生成一次性的上传签名然后将签名交给 App。因为服务器一般很难被攻陷，所以安全性是可以保证的。
- 发布短视频时，请务必保证正确传递 Signature 字段，否则会发布失败；

对接攻略



1、选择视频

可以接着上几篇文档中的录制或者编辑或者拼接，把生成的视频进行上传，也可以选择手机本地的视频进行上传

2、压缩视频

- 压缩视频会减小视频大小，但是也会降低视频的清晰度，您可以根据自己的需求决定是否进行压缩。
- 对选择的视频进行压缩，使用TXVideoEditor.generateVideo(int videoCompressed, String videoOutputPath)接口，支持4种分辨率的压缩，后续会增加自定义码率的压缩。

3、发布视频

把刚才生成的 MP4 文件发布到腾讯云上，App 需要拿到上传文件用的短期有效上传签名，这部分有独立的文档介绍，详情请参考 [签名派发](#)。

TXUGCPublish (位于 TXUGCPublish.java) 负责将 MP4 文件发布到腾讯云视频分发平台上，以确保视频观看的就近调度、秒开播放、动态加速 以及海外接入等需求。

```
mVideoPublish = new TXUGCPublish(TCVideoPublisherActivity.this.getApplicationContext());  
// 文件发布默认是采用断点续传  
TXUGCPublishTypeDef.TXPublishParam param = new TXUGCPublishTypeDef.TXPublishParam();  
param.signature = mCosSignature; // 需要填写第四步中计算的上传签名  
// 录制生成的视频文件路径, ITXVideoRecordListener 的 onRecordComplete 回调中可以获取  
param.videoPath = mVideoPath;  
// 录制生成的视频首帧预览图, ITXVideoRecordListener 的 onRecordComplete 回调中可以获取  
param.coverPath = mCoverPath;  
mVideoPublish.publishVideo(param);
```

发布的过程和结果是通过 TXRecordCommon.ITXVideoPublishListener (位于 TXRecordCommon.java 头文件中定义) 接口反馈出来的 :

- onPublishProgress 用于反馈文件发布的进度 , 参数 uploadBytes 表示已经上传的字节数 , 参数 totalBytes 表示需要上传的总字节数。

```
void onPublishProgress(long uploadBytes, long totalBytes);
```

- onPublishComplete 用于反馈发布结果 , TXPublishResult 的字段 errCode 和 descMsg 分别表示错误码和错误描述信息 , videoURL 表示短视频的点播地址 , coverURL 表示视频封面的云存储地址 , videoId 表示视频文件云存储 Id , 您可以通过这个Id调用点播 [服务端 API 接口](#)。

```
void onPublishComplete(TXPublishResult result);
```

- 通过 [错误码表](#) 来确认短视频发布的结果。

4、播放视频

- 第 3 步上传成功后 , 会返回视频的 fileId , 播放地址 url , 封面 url。用 [点播播放器](#) 可以直接传入fileId播放 , 或者url播放。