

# 趣看云视频直播点播服务 录播上传SDK iOS开发指南

□□□1.0

2016.05



## SDK 概述

录播上传 SDK 是趣看官方推出的录播软件开发工具包，其中 iOS SDK 为 iOS 开发者提供简单、快捷的接口，帮助开发者在 iOS 平台上快速开发本地视频应用。

SDK 支持将本地视频文件，上传存储到阿里云中，并且进行视频转码 以及 视频文件合并的操作。

SDK 支持 iOS 7.0 及以上版本。

由于一次录播时间有可能会很长，所以 sdk 对录像的录制进行了分片处理，目前设定为 5 分钟生成一个 mp4 的录像文件。

用户将这些 MP4 文件上传到阿里云服务器后，可以通知我们的服务器，对这些文件进行文件合并，最终可以在服务器上生成一个完整的视频文件。

Demo 的作用就是在本地建立一个数据库，管理这些文件及上传状态。

## 开发准备

- 在趣看后台生成账号，获取到账号对应的 APPKEY
- 下载录播上传 iOS SDK
- 解压下载的录播上传 SDK 开发包，将头文件和静态库都导入工程。
- 添加 SDK 依赖库
  - libz.dylib
  - AVFoundation.framework
- 在 Other Linker Flags 中添加 -ObjC
- 使用 SDK 前，要先调用[IPLocalCameraSDK initWithAppkey:]设置 Appkey。
- 在需要使用录播的地方 import IPLocalCameraSDK.h。
- 在需要使用上传的地方 import IPUploadVideo.h。

## SDK 中录制视频类说明

SDK 中与录播有关的功能都封装在 IPLocalCameraSDK 类中，下面详细介绍下 IPLocalCameraSDK 类的功能。

## 初始化SDK，并设置日志信息

```
+(void)initSdkSaveLog:(int)saveLog;
```

此接口初始化 IPLocalCameraSDK 接口，并配置了日志的显示情况。

## 配置获取录播完成后的文件名

```
+(void)getFileEndRecord:(getFileEndRecord)endrecord;
```

此接口在录播过程中，每个录播分片完成后都会回调 `getFileEndRecord` 通知用户这个分片录播完成。`getFileEndRecord` 是一个 block 回调。

## 录播参数设置

IPLocalCameraSDK 中录播相关的参数为：视频的清晰度、视频的码率、视频的帧率：

- 设置视频清晰度

```
+(void)setLocalCameraEncodeType:(NSInteger)encodeType;
```

- 设置视频码率

```
+(void)setLocalCameraVideoBitRate:(NSInteger)videobitrate;
```

- 设置视频帧率

```
+(BOOL)setVideoFrame:(int)videobitrate;
```

## 摄像头开关 ( startCamera与stopCamera )

IPLocalCameraSDK 提供用户开启和关闭摄像头的操作：

- 录播前开启摄像头

```
+(int) startLocalCameraNoEncodeType:(UIView *)view VideoPath:(NSString*) videoPath
GainValue:(float)gain_value Orientation:(AVCaptureVideoOrientation)Orientation
Camera:(NSInteger)cameraValue;
```

view: 摄像头采集到的画面, 所显示的画布。  
VideoPath: 选录像保存的文件夹的在 documents 下的相对路径。  
GainValue: 声音增大效果 范围 0 - 1。  
Orientation: 竖屏拍摄还是横屏拍摄, AVCaptureVideoOrientationPortrait 竖拍  
AVCaptureVideoOrientationLandscapeRight 横拍。  
CameraValue:摄像头。0: 前置摄像头 1、后置摄像头。

- 录播完成后关闭摄像头

```
-(void)stopLocalCamera;
```

## 开始和停止录播

IPLocalCameraSDK 提供录播开启和停止的接口, 方便用户控制那些视频需要进行录制, 那些不需要:

- 开始录播

```
+(int) startLocalRecord;
```

0 正常 -1 启动非正常 -2APPKEY 验证未能正常进行

- 停止录播

```
+(int) stopLocalRecord;
```

停止录播后, 只要摄像头没关闭, 还可以调用 startLocalRecord, 重新开启录播。

## 开启或者关闭音频

IPLocalCameraSDK 在开启摄像头后, 提供任意时刻的开关音频的功能:

```
+(void) switchLocalAudio :(int) audioType;
```

参数 0 关闭音频, 1 开启音频。

## 录播中修改声音增益的效果

IPLocalCameraSDK 在开启摄像头后，提供任意时刻的修改声音增益属性：

```
+(void) changeLocalGainValue : (float)gain_value;
```

参数 gain\_value 范围 0-1，声音增益的幅度，0 最小 1 最大。

## 切换摄像头

IPLocalCameraSDK 在开启摄像头后，提供在任意时刻的前后摄像头切换功能：

```
+(BOOL) switchLocalChangeCamera : (int) cameraValue;
```

返回切换摄像头成功或者失败，参数 0：启动前置摄像头 1：启用后置摄像头。

## 摄像头闪光灯的开启和关闭

IPLocalCameraSDK 在开启摄像头后，可以开启或者关闭闪光灯。

```
+(BOOL) switchLocalFlash:(int)flashFlag;
```

返回开启闪光灯成功或者失败，参数： 0：关闭闪光灯。 1：开启闪光灯。

## 摄像头聚焦模式改变

IPLocalCameraSDK 在开启摄像头后，可以改变其聚焦模式。

```
+(BOOL) switchLocalFocusMode : (int) focusType;
```

返回改变聚焦模式成功或者失败，参数： 0：自动聚焦。 1：手动聚焦。

- 手动聚焦设置焦点

```
+(void) manualLocalFocus:(CGPoint) touchPoint TouchView:(UIView *)view;
```

在调用 switchLocalFocusMode(1)后，切换到手动聚焦的模式，就可以调用该函数使手机的摄像头修改聚焦的焦点位置。

## 摄像头焦距改变

IPLocalCameraSDK 在开启摄像头并且你的设备系统高于 IOS7,设备型号为 Iphone5 及以上设备,你可以在任意时刻实时改变摄像头的焦距:

```
+(void) zoomLocalOut : (float) scale;
```

在开启摄像头后,就可以设置摄像机焦距,参数范围: 1-10, 1 物体正常大小 10 物体被放大多倍。

## 设置和清除水印

IPLocalCameraSDK 提供设置水印的接口,用于给视频添加 Logo,在初始化 SDK 后可以调用:

- 设置水印

```
+(void)setLocalUserImage:(UIImage*)logoimg Transparent:(int)transparent Offset:(int) offset;
```

参数 transparent logo 图片的透明度 0-10 之间, 0 完全透明, 10 不透明

参数 offset logo 图片的位置 0 表示设置 logo 在左上角 1 表示设置 logo 在右上角。每次设置 logo 前,先调用 clearUserImage 清除已有 logo。Logo 要针对分辨率进行设置,其中,所有 16:9 的图像 logo 标准按照 1280\*720 来的,640\*480 与 352\*288 的 logo 需单独设置。

- 清除水印

```
+(void)clearLocalUserImage;
```

清除已有 logo 水印。

## 录播sdk版本号

IPLocalCameraSDK 提供接口,获取当前 sdk 版本号:

```
+(NSString*)getVersion;
```

## SDK 中上传文件类说明

SDK 中与文件上传有关的功能都封装在 `IPUploadVideo` 类中，下面详细介绍下 `IPUploadVideo` 类的功能。

使用 `IPUploadVideo` 功能前，请先在 `IPLocalCameraSDK` 中设置 `Appkey`。

### 设置文件上传中的状态回调

```
[IPUploadVideo changeBlock:^(UploadedState now, id data) {}];
```

设置回调用于监听文件上传中出现的各种问题或者状态。`UploadedState` 为各种状态的枚举，`data` 为返回的数据，具体状态与返回数据，请参考 `api` 文档。

### 上传视频文件对象

`IPUploadVideo` 中需要对想要上传的视频文件分装成对象 `QKRecord`，再调用上传接口上传：

- `QKRecord` 对象属性介绍

```
NSNumber *liveid; //这个录像文件的唯一标示
```

```
NSString *fileName; //在后台显示的文件名,文件名规则为：唯一标示符_第几个分片.后缀，同一
```

批录播文件的唯一标示符必须相同，例如：`aaa_1.mp4`, `aaa_2.mp4`, 代表的是标识符为 `aaa` 的第

一个和第二个录像分片，唯一标识符要尽量复杂，确保唯一。如果是本地视频也要参照这个规则，

就是只有一个分片的录像

```
NSString *uploadfileName; //在阿里云服务器上的文件名
```

NSString \*address; //文件在 documents 目录下的相对路径,格式为: 文件夹/文件名,例如:上传在

documents 的 Record 目录下的 mp\_1.mp4 文件,路径填写:Record/mp\_1.mp4

NSString \*type; //文件类型 localvideo:录播视频,目前支持一种 type

- 上传视频

```
StartUpload state = [IPUploadVideo startLoadRecordData:data progress:^(double progress,NSNumber *liveId) {}];
```

上传视频文件,并且设置文件进度会掉的 block。Progress 是百分数,表示上传的进度, liveId 是文件唯一标示。

## 暂停再上传的文件

```
+(void)cancelUploading:(NSNumber*)liveid allcancel:(BOOL)allcancel;
```

该函数用来暂停上传进度,若是 allcancel 填 YES 的时候,暂停此次上传。设置为 NO 的时候,会去匹配 liveid,匹配上了则暂停,匹配不上还是会继续上传。

## 删除某次上传

```
+(void)deleteUploading:(QKRecord*)data;
```

该函数用来删除本次上传的进度,清除服务器的断点续传的缓存,以及本地上传的记录,对已经上传完成的文件无效。

## 通知服务器上传完成

文件上传到阿里云后,需要通知服务器该文件上传已完成,SDK 会自动通知服务器文件上传完成,但是有时会碰到通知失败的情况,用户可以在接受到失败信息后,手动再次通知。

```
[IPUploadVideo noticeServerUploaded:data noticeServer:^(UploadedState now,id data){
}];
```

该函数用来重新通知服务器上传完成了



## 通知服务器合并一系列文件

文件上传到阿里云后，需要通知服务器该文件上传已完成，SDK 会自动通知服务器文件上传完成，但是有时会碰到通知失败的情况，用户可以在接受到失败信息后，手动再次通知。

```
[IPUploadVideo uploadVideoMerger:array noticeServer:^(UploadedState now,id data){
}];
```

该函数用来重新通知服务器，合并一系列已经上传完成的文件，文件名必须按录像顺序排列，文件名开头的唯一标识符必须相同，文件名格式必须为：唯一标识\_第几个分片.文件后缀名，如：aaa\_2.mp4。

## 检查正在上传的文件

该函数用来获取当前正在上传的文件的唯一标示，没有则返回 nil

```
+(NSNumber*)checkNowRecord;
```

返回值为 QKRecord 对象中的 liveId，表示该文件的唯一标识。

---

## API文档

IPCameraSDK.h

```
+(int)initAppkey:(NSString*)appkey; 初始化 appkey
```

```
+(BOOL) setLocalCameraEncodeType:(NSInteger)encodeType; 设置视频清晰度
```

```
+(BOOL) setLocalCameraVideoBitRate:(NSInteger)videobitrate; 设置视频码率
```

```
+(BOOL) setVideoFrame:(int)videoframe; 设置帧率
```

+(int) startLocalCameraNoEncodeType:(UIView \*)view VideoPath:(NSString\*) videoPath  
GainValue:(float)gain\_value Orientation:(AVCaptureVideoOrientation)Orientation

Camera:(NSInteger)cameraValue; 开启摄像头

+(void) stopLocalCamera; 关闭摄像头

+(int) startLocalRecord; 0 正常 -1 启动非正常 -2APPKEY 验证未能正常进行

+(void) stopLocalRecord;停止录播后，只要摄像头没关闭，还可以调用 startLocalRecord，重新开启录播。；关闭 rtmp 发送

+(void)setLocalUserImage:(UIImage\*)logoimg Transparent:(int)transparent Offset:(int) offset; 添加 logo，transparent 为 logo 所占的透明度，参数 0-10，offset 目前支持 kImageLeftOffset 左边和 kImageRightOffset 右边

+(void)clearLocalUserImage; 清空所有 logo

+(BOOL) switchLocalFlash:(int)flashFlag; 开启或关闭闪光灯：0 关闭 1 开启

+(void) switchLocalAudio :(int) audioType; 音频开关：0 关闭音频，1 打开音频

+(BOOL) switchLocalChangeCamera :(int) cameraValue; 摄像头切换：0 前置摄像头，1 后置摄像头

+(BOOL) switchLocalFocusMode : (int) focusType; 设置聚焦 : 0 自动聚焦 , 1 手动聚焦

+(void) manualLocalFocus:(CGPoint) touchPoint TouchView:(UIView \*)view; 手动聚焦的聚焦点

+(void) zoomLocalOut : (float) scale; 摄像头焦距设置

+(void) changeLocalGainValue : (float)gain\_value; 声音增益 , 参数 float 0-1

+(NSString\*)getVersion; 获取版本号

## IPUploadVideo.h

[IPUploadVideo changeBlock:^(UploadedState now, id data) {}]; 设置文件上传中的状态回调

StartUpload state = [IPUploadVideo startLoadRecordData:data progress:^(double progress,NSNumber \*liveId) {}]; 上传视频文件对象

+(void)cancelUploading:(NSNumber\*)liveid allcancel:(BOOL)allcancel; 暂停再上传的文件

+(void)deleteUploading:(QKRecord\*)data; 删除某次上传

[IPUploadVideo noticeServerUploaded:data noticeServer:^(UploadedState now,id data){

}); 通知服务器上传完成

```
[IPUploadVideo uploadVideoMerger:array noticeServer:^(UploadedState now,id data){
```

```
}}]; 通知服务器合并一系列文件
```

```
+(NSNumber*)checkNowRecord; 检查正在上传的文件
```

## QKRecord.h

```
NSNumber *liveid; //这个录像文件的唯一标示
```

```
NSString *fileName; //在后台显示的文件名,文件名规则为：唯一标示符_第几个分片.后缀，同一批录播文件的唯一标示符必须相同，例如：aaa_1.mp4, aaa_2.mp4,代表的是标识符为 aaa 的第一个和第二个录像分片，唯一标识符要尽量复杂，确保唯一。如果是本地视频也要参照这个规则，就是只有一个分片的录像。
```

```
NSString *uploadfileName; //在阿里云服务器上的文件名
```

```
NSString *address; //文件在 documents 目录下的相对路径,格式为：文件夹/文件名，例如：上传在 documents 的 Record 目录下的 mp_1.mp4 文件，路径填写:Record/mp_1.mp4
```

```
NSString *type; //文件类型 localvideo:录播视频，目前支持一种 type
```