

阿里巴巴集团

SDK 参考手册

iOS 媒体推流器 SDK 参考手册

POWERED BY QUPAI

文档修订历史

版本	作者	工作描述	修订历史	修改日期
1.0	刘亚钊	iOS 推流 SDK 文档		2016-06-21

1.1	刘亚钊	iOS 推流 SDK 文档		2016-07-14

文档说明

本文档面向所有使用该 SDK 的开发人员、测试人员以及对此感兴趣的用
户，要求读者具有一定的 iOS 开发能力

1. 简介

iOS 推流 SDK 是在 iOS 平台上使用的软件开发工具包(Soft Development Kit)，为
iOS 开发者提供简单易用的接口，帮助开发者实现 iOS 平台上的推流应用开发。

1.1 功能说明

- 1. 方便快捷、低门槛实现媒体推流功能。用户无须关心内部实现细节，只需要
自定义界面既可以实现专业级的推流应用。

2. 推流支持格式：rtmp

3. 编码目前为硬编

1.2 安装包说明

推流器 SDK 的完整下载包中包含 test、lib、doc 等：

1. test：主要存放了调用 SDK 的示例工程，可以帮助用户了解如何使用该 SDK

2. lib：推流器 SDK 开发包, 包含推流器 framework 文件, 需要在您的工程
中进行引用

3. doc：存放 SDK 相关接入文档。

1.3 推流器性能

1. 目前推流 SDK 推流采用的是硬编。

2. 推流采用 librtmp 推流

3. SDK 的大小：使用 SDK 库安装之后的程序大小在 1M 左右。即为你的程序增加 1M 左右的大小。

1.4 注意事项

1. 推流器 SDK 目前只支持单实例。不能够同时开 2 个推流实例，同时只能存在一个实例，需要另开实例的时候，需要关闭之前存在的实例。

2. 操作系统版本要求 iOS8.0 以上。

2. 使用说明

2.1 开发环境配置

1. 需要准备 iOS 的运行环境(XCode6.0 以上版本, iOS SDK8.0 以上版本), 以及硬件 CPU 支持 ARMv7、ARMv7s 或 ARM64 的 iOS 设备。

2.2、开通直播

参考链接:

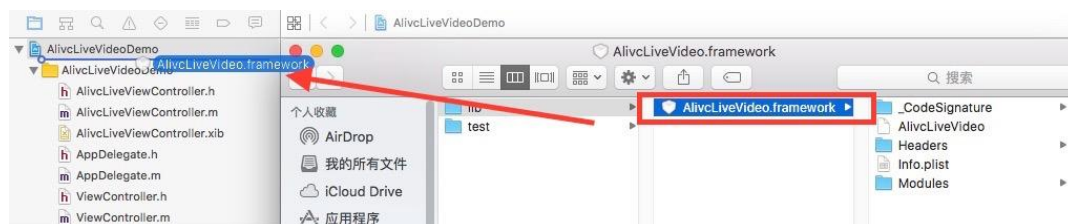
https://help.aliyun.com/document_detail/29957.html

2.3、SDK 下载

—阿里云提供下载地址。

2.4、SDK 集成

拖拽 sdk 文件夹到自己的 Xcode 项目中。



勾选 Copy items if needed, 点击 Finish。

打开项目的 app target, 在 Build Phases 中的 Link Binary With Libraries 添加以下依赖库。

```
libz.tbd  
VideoToolbox.framework  
AudioToolbox.framework  
libstdc++.tbd  
SystemConfiguration.framework  
CoreTelephony.framework
```

2.5、直播功能的使用

在需要直播功能的控制器头部导入#import
<AlivcLiveVideo/AlivcLiveVideo.h>, 并设置代理
<AlivcLiveSessionDelegate>

1.设置直播参数

```
AlivcLConfiguration *configuration = [[AlivcLConfiguration alloc] init];  
configuration.url = pushUrl;//设置推流地址  
configuration.videoMaxBitRate = 1500 * 1000;//设置最大码率，设置最大码率和  
最小码率后 SDK 会根据网络状况自动调整码率  
configuration.videoBitRate = 600 * 1000;//设置当前视频码率  
configuration.videoMinBitRate = 400 * 1000;//设置最小码率  
configuration.audioBitRate = 64 * 1000;//设置音频码率  
configuration.videoSize = CGSizeMake(360, 640);//设置直播分辨率  
configuration.screenOrientation = AlivcLiveScreenVertical;// 设置横屏 or 竖屏  
configuration.fps = 20;//设置帧数  
configuration.preset = AVCaptureSessionPresetiFrame1280x720;//设置采集质量  
configuration.position = AVCaptureDevicePositionFront;//设置前置摄像头或后置  
摄像头
```

2.创建直播 session

```
_liveSession = [[AlivcLiveSession alloc] initWithConfiguration:configuration]
```

3.设置 session 代理

```
_liveSession.delegate = self;
```

4.开启直播预览

```
[_liveSession alivcLiveVideoStartPreview];
```

5.开启直播

```
[_liveSession alivcLiveVideoConnectServer];
```

6.获取直播预览视图

```
[_liveSession previewView]
```

可将获取的直播视图添加到当前控制器上，实现直播预览功能。

7.停止预览

```
[_liveSession alivcLiveVideoStopPreview];
```

注意：停止预览后将 liveSession 置为 nil

8.关闭直播

```
[_liveSession aliveLiveVideoDisconnectServer];
```

9.设置闪光灯模式

```
_liveSession.torchMode = AVCaptureTorchModeOn//关闭  
AVCaptureTorchModeOff
```

10.开启美颜

```
[_liveSession setEnableSkin:YES];
```

11.缩放

```
[_liveSession aliveLiveVideoZoomCamera:1.0f];
```

12.聚焦

```
[_liveSession aliveLiveVideoFocusAtAdjustedPoint:percentPoint autoFocus:YES];
```

13. 调试信息

```
AliveLDebugInfo *i = [_liveSession dumpDebugInfo];
```

2. 6、直播功能的相关代理方法

@optional

1. 直播出错的代理方法

```
-(void)aliveLiveVideoLiveSession:(AliveLiveSession *)session  
error:(NSError *)error;
```

2. 网速较慢时的代理方法

```
-(void)aliveLiveVideoLiveSessionNetworkSlow:  
(AliveLiveSession *)session;
```

@required

1. 音频初始化失败

```
-(void)alivcLiveVideoLiveSession:(AlivcLiveSession *)session  
OpenAudioError:(NSError *)error;
```

2. 视频初始化失败

```
-(void)alivcLiveVideoLiveSession:(AlivcLiveSession *)session  
OpenVideoError:(NSError *)error;
```

3. 音频编码器初始化失败

```
-(void)alivcLiveVideoLiveSession:(AlivcLiveSession *)session  
EncodeAudioError:(NSError *)error;
```

4. 视频编码器初始化失败

```
-(void)alivcLiveVideoLiveSession:(AlivcLiveSession *)session  
EncodeVideoError:(NSError *)error;
```

2.7、调试信息说明

- cameraPresent	摄像头方向
- connectStatus	连接状态
- fps	当前编码帧数
- encodeSpeed	编码速度
- speed	当前上传速度 单位byte
- localBufferSize	本地buffer大小
- localBufferAudioCount	本地buffer音频帧数
- localBufferVideoCount	当前buffer视频帧数
- localDelay	编码耗时 单位ms
- pushSize	当前上传数据大小 单位 byte
- keyFrameDTS	上一个关键帧dts
- currentVideoPTS	当前输出流video pts
- currentAudioPTS	当前输出流audio pts
- encodeFrameCount	所有编码帧数
- pushFrameCount	所有发送帧数
- videoDiscardFrameCount	视频丢帧数
- audioDiscardFrameCount	音频丢帧数
- cycleDelay	周期性延迟 单位 ms

- eventArray

事件信息数组，内部数据结构为打点事件的字典

3. 注意事项

无

4. 版权声明

版权所有，切勿盗版