

iOS UPhone – **SDK** 接入文档

变更历史:

SDK 版本	发版日期	更新内容
1.0.0	2021/11/26	1、初始版本
1.0.1	2022/01/05	1、新增开始直播、停止直播接口 2、新增获取当前播放器截屏接口 3、新增检测当前截屏是否黑屏接口 4、新增获取丢包率接口 5、新增获取视频流基本参数接口 6、新增获取用户最后一次操作时间戳接口 7、新增是否支持直播接口 8、新增开启静音以及关闭静音接口
1.0.2	2022/01/19	1、优化初始化SDK逻辑 2、优化连接云手机逻辑
1.0.3	2022/03/25	1、优化声音问题 2、根据视频流大小横竖屏转换、根据第一帧返回比例调整切换分辨率的传值
1.0.4	2022/06/10	1、新增云手机清理后台任务 2、优化设置分辨率

目录

目录

1	工程配置.....	4
1.1	导入SDK.....	4
1.2	配置权限.....	4
1.2.1	Target配置.....	4
1.2.2	Info.plist文件配置.....	4
2	接入步骤.....	4
2.1	初始化 SDK.....	4
3	接口说明.....	5
3.1	连接云手机.....	5
3.2	断开云手机.....	5
3.3	重连云手机.....	5
3.4	设置分辨率.....	6
3.5	获取网络延时.....	6
3.6	获取SDK版本号.....	7
3.7	云手机清理后台任务.....	7
3.8	云手机返回到桌面.....	7
3.9	云手机返回到上一级界面.....	8
3.10	获取网络速度.....	8
3.11	开始直播.....	8
3.12	停止直播.....	9
3.13	获取当前播放器截屏.....	9
3.14	检测当前截屏是否黑屏.....	9
3.15	获取丢包率.....	10
3.16	获取视频流基本参数接口.....	10
3.17	获取用户最后一次操作时间戳.....	10
3.18	是否支持直播.....	11
3.19	静音开关功能.....	11
4	注意事项.....	12
5	官方Demo.....	12

1 工程配置

欢迎使用云游戏 SDK 。为方便 iOS 开发者调试，这里向您介绍适用于开发的工程配置。

1.1 导入 SDK

将我们提供的SDK 压缩包中 UPhoneSDK.framework 导入项目中（记得选上Copy items if needed）。

1.2 配置权限

1.2.1 Target 配置

Target中所需配置如下：

- 1、 TARGET -> General -> Frameworks,Libraries,and Embedded Contend -> UPhoneSDK.framework 选择 Embed & Sign.
- 2、 TARGET->Build Settings -> Build Options-> Enable Bitcode 选择 NO.

1.2.2 Info.plist 文件配置

在工程文件下的 Info.plist文件中加入以下代码：

```
<key>NSMicrophoneUsageDescription</key>
<string>此App将要访问您的麦克风</string>
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
```

注：麦克风权限文字可根据自己App修改。

2 接入步骤

为方便 iOS 开发者调试和接入云游戏产品 API，这里向您介绍适用于 iOS 开发的快速接入文档。快速入门文档只提供最主要的接入接口，更多详细接口请参考 本文第 4 章节“接口说明”部分。

重要接口	接口含义	建议调用时机
initWithUphone	初始化SDK	连接云手机需要展示云手机界面时

说明：

- SDK 使用前请对工程进行配置，否则 SDK 不生效。

2.1 初始化 SDK

函数原型

```
-(instancetype)initWithUphone:(NSString *)uphoneId;
```

参数	类型	意义
uphoneId	NSString	接入商的唯一 ID，用来区分不同的接入商。

示例代码

```
UINavigationController *videoCallViewController =  
[[UINavigationController alloc] initWithUphone:uphoneId];
```

3 接口说明

3.1 连接云手机

函数原型

```
-(void)connectUPhone;
```

示例代码

```
[self connectUPhone];
```

注：self是UPhoneVideoViewController的子类

3.2 断开云手机

函数原型

```
-(void)disconnectUPhone;
```

示例代码

```
[self disconnectUPhone];
```

注：self是UPhoneVideoViewController的子类

3.3 重连云手机

重新连接云手机

函数原型

```
+ (void)reconnetUPhone;
```

示例代码

```
[UPhoneService reconnetUPhone];
```

3.4 设置分辨率

设置UPhone分辨率

函数原型

```
- (void)setUPhoneResolution:(int)resolution;
```

参数	类型	意义
resolution	int	0// 标清 3 // 高清 6// 超清

示例代码

```
[self setUPhoneResolution: resolution];
```

注：self是UPhoneVideoViewController的子类

3.5 获取网络延时

获取云手机网络延时

函数原型

```
- (NSInteger)getUPhoneLinkDelay;
```

返回参数说明： 返回值类型为NSInteger，单位为ms。

示例代码

```
Integer delay = [self getUPhoneLinkDelay];
```

注：self是UPhoneVideoViewController的子类。

3.6 获取SDK版本号

获取版本号

函数原型

```
+ (NSString *) getVersionCode;
```

示例代码

```
NSString *delay = [UPhoneService getVersionCode];
```

3.7 云手机清理后台任务

云手机清理后台任务

函数原型

```
- (void)backUPhoneTaskList;
```

示例代码

```
[self backUPhoneTaskList];
```

注：self是UPhoneVideoViewController的子类

3.8 云手机返回到桌面

云手机返回到桌面

函数原型

```
- (void)backUPhoneHome;
```

示例代码

```
[self backUPhoneHome];
```

注：self是UPhoneVideoViewController的子类

3.9 云手机返回到上一级界面

云手机返回到上一级界面

函数原型

```
- (void)backUPhoneLastPage;
```

示例代码

```
[self backUPhoneLastPage];
```

注：self是UPhoneVideoViewController的子类

3.10 获取网络速度

云手机获取当前网络速度

函数原型

```
-(NSString *)getNetworkSpeed;
```

返回参数说明：返回值类型为NSString，单位为 MB/s，保留两位小数，例如：0.15MB/s。

示例代码

```
NSString *networkSpeed = [self getNetworkSpeed];
```

注：self是UPhoneVideoViewController的子类

3.11 开始直播

开始直播推流

函数原型

```
-(void)startLiving:(NSString *)url ;
```

参数	类型	意义
url	NSString	直播推流地址

示例代码

```
[strongSelf startLiving:url];
```


注：self是UPhoneVideoViewController的子类

3.12 停止直播

停止直播推流

函数原型

```
-(void)stopLiving;
```

示例代码

```
[self stopLiving];
```

注：self是UPhoneVideoViewController的子类

3.13 获取当前播放器截屏

云手机启动后，如果想要获取当前播放器截屏，可以调用此函数

函数原型

```
-(UIImage *)getShortcut;
```

返回参数说明： 返回当前播放器截屏的 UIImage 对象

示例代码

```
UIImage *image = [self getShortcut];
```

注：self是UPhoneVideoViewController的子类

3.14 检测当前截屏是否黑屏

如果检测当前截屏的UIImage对象是否黑屏功能，调用此函数

函数原型

```
-(NSInteger)checkBlackScreen:(UIImage *)image;
```

参数	类型	意义
image	Image	当前需要检测的UIImage对象

返回参数说明：

1//检测结果为纯色 排除黑色和全透明色，因为播放器没开始工作，不通设备获取截屏有的是纯黑色有的是纯透明色

2//检测结果为纯透明色

3//检测结果为正常

4//检测结果为纯黑色

示例代码

```
NSInteger color = [self checkBlackScreen:image];
```

注：self是UPhoneVideoViewController的子类

3.15 获取丢包率

获取网络传输过程中的丢包率

函数原型

```
-(NSString *)getLossRate;
```

返回参数说明：返回值类型为NSString，已转化成百分比并且保留两位小数，例如：0.15%。

示例代码

```
NSString *lossRate = [self getLossRate];
```

注：self是UPhoneVideoViewController的子类。

3.16 获取视频流基本参数接口

获取视频分辨率、横竖屏参数

函数原型

```
-(NSDictionary *)getQRCodeData;
```

返回参数说明:NSDictionary 类型参数

// key值为style, value返回值为0时表示横屏, value为1时表示竖屏;

// key值为height, value返回值即为当前分辨率的height;

// key值为width, value返回值即为当前分辨率的width。

示例代码

```
NSDictionary *dic = [strongSelf getQRCodeData];
NSString *style = [dic valueForKey:@"style"];
NSString *height = [dic valueForKey:@"height"];
NSString *width = [dic valueForKey:@"width"];
```

注：self是UPhoneVideoViewController的子类。

3.17 获取用户最后一次操作时间戳

云手机启动后，通过该接口获取用户最后一次操作实例的时间戳

函数原型

```
+ (NSString *)getLastUserOperationTimestamp;
```

返回参数说明:NSString 类型参数，单位为：ms。

0//默认返回 0，代表用户没有操作过实例，否则返回相应时间戳

示例代码

```
NSString *lastUserOperationTimestamp = [UPhoneService getLastUserOperationTimestamp];
```

3.18 是否支持直播

云手机启动后，通过该接口获取是否支持直播

函数原型

```
- (NSString *)isSupportLiving;
```

返回参数说明:NSString 类型参数

0//未设置

1//正在启动推流

2//正在推流

3//已停止推流

示例代码

```
NSString *isLiveStr = [self isSupportLiving];
```

注：self是UPhoneVideoViewController的子类

3.19 静音开关功能

设置当前的播放是否为静音状态

函数原型

```
- (void)setAudioMute:(BOOL)isMute;
```

传入参数说明:BOOL类型参数

YES：全局静音

NO：取消全局静音

示例代码

```
BOOL mute1 = [self setAudioMute:YES];  
BOOL mute2 = [self setAudioMute:NO];
```

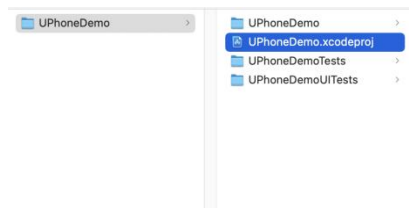
注：self是UPhoneVideoViewController的子类

4 注意事项

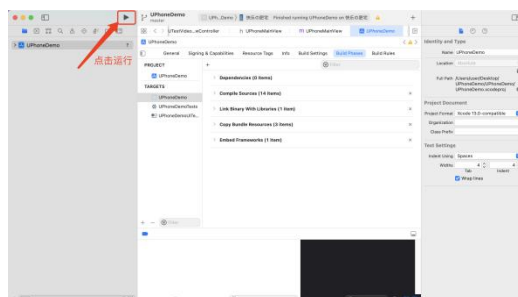
1. 该SDK仅支持iOS10以上系统。
2. 该SDK仅支持真机运行。
3. 需要用到SDK的地方增加头文件#import <UPhoneSDK/UPhoneSDK.h>
4. 跳转到新建子类的界面时，前者需要遵守<UPhoneVideoViewControllerDelegate>协议，在退出云手机时需要的一些方法可以写在协议方法里面。
5. 每次进入云手机会从远端获取分辨率，可以根据自己的需求修改相应的分辨率可以参照[3.4 设置分辨率](#)。

5 官方 Demo

1. 下载官方 iOS 端 UPhoneDemo 工程文件。
2. 双击打开UPhoneDemo.xcodeproj 文件进入Xcode。



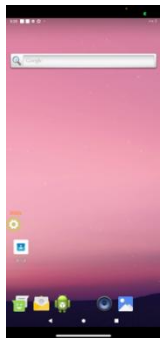
3. 运行UPhoneDemo代码



4. 启动安装好的 App，输入云手机Id，进入云手机界面。



5. 出现云手机画面，则表示云手机 Demo 运行成功。



6. 如果想要退出云手机，则选择设置面板中退出云手机即可。

