

GotyeLiveSDK 使用指南

SDK 概述

GotyeLiveSDK 是一套完整的视频直播 SDK 产品，含有以下四个模块：

- **GotyeLiveCore** 亲加视频直播 SDK 的基础模块，包含了各个模块所必需的接口实现，在集成其它模块之前都需要先把 GotyeLiveCore SDK 集成到项目中。
- **GotyeLiveChat** 聊天模块。实现了用户在聊天室内收发消息的功能。
- **GotyeLivePlayer** 播放器模块。实现了在聊天室内观看当前直播视频的功能。
- **GotyeLivePublisher** 视频发布模块。实现了主播账号在聊天室内直播视频的功能。

除了 GotyeLiveCore 模块是必须的之外，其余模块开发者可以根据实际需求进行选择。文档接下来的部分将会对各个模块的进行详细说明。

GotyeLiveCore

集成步骤

1. 在 GotyeLiveCore 文件夹中，有一个 libGotyeLiveCore.a 文件和 include、third-party 文件夹，将它们导入工程。其中，third-party 文件夹中为 SDK 中用到的一些第三方开源库，如果工程中也使用了同样的第三方库，建议使用 SDK 中包含的版本。
2. 添加依赖库
 - MobileCoreServices.framework
 - SystemConfiguration.framework
 - libicucore.tbd（XCode6 及以下版本后缀名为 dylib）
3. 在 Other Linker Flags 中添加 -ObjC
4. 在工程配置 Info.plist 文件中，添加条目 NSAppTransportSecurity，属性设

为 Dictionary。在此条目下添加键值对，key 为 NSAllowsArbitraryLoads, 值为 YES。

初始化 SDK

1. 在需要使用地方导入头文件 GLCore.h
2. 添加 SDK 初始化方法

```
[[GLCore sharedInstance] initWithAccessSecret:你的认证 key  
                                             companyId:你的公司的唯一标示];
```

验证房间 ID 和密码

调用

```
[[GLCore sharedInstance] authRoomId:房间号  
                                password:房间密码  
                                bindAccount:绑定账号  
                                nickname:用户昵称  
                                isCustomRoomId:是否第三方房间号  
                                success:^(GLAuthToken *authToken) {  
                                    //认证成功回调，authToken为具体的认证信息  
                                } failure:^(NSError *error) {  
                                    //认证失败回调，error为具体的错误信息  
                                }];
```

注：如无特别说明，回调的 block 都是在主线程中运行的。

清除验证消息

在用户退出房间后，应该调用

```
[[GLCore sharedInstance] clearAuthorization];
```

GotyeLiveChat

集成步骤

1. 集成 GotyeLiveCore（如已集成略过这一步）。
2. 将 libGotyeLiveChat.a 以及同级目录下的 include 文件夹导入到工程中

初始化 SDK

1. 在需要使用的地方导入头文件 GLChatManager.h
2. 加 SDK 初始化方法

```
[GLChatKit initWithSession: GLCoreKit.roomSession];
```

登录聊天室

调用

```
[GLChatKit enterRoomOnSuccess:^( NSString *account, NSString *nickname) {  
    //登录成功回调  
} failure:^(NSError *error) {  
    //登录失败回调  
}];
```

接收/发送消息

登录成功之后，可以在聊天室中发送消息

```
[GLChatKit sendText:发送的文本内容  
    extra:附加内容  
    success:^( ) {  
        //发送成功回调  
    }  
    failure:^(NSError *error) {  
        //发送失败回调  
    }];
```

添加监听器后，当收到新的消息时，能通过回调获取新的消息

```
//添加监听器  
[GLChatKit addObserver:self];  
  
- (void)chatClientDidReceiveMessage:(GLChatMessage *)msg  
room:(NSString *)roomId  
{  
    //收到消息回调成功回调  
}
```

退出聊天室

需要退出登录时，可调用

```
[GLChatKit leaveRoom];
```

GotyeLivePlayer

集成步骤

1. 集成 GotyeLiveCore（如已集成略过这一步）。
2. 将 libGotyeLivePlayer.a 以及同级目录下的 include 文件夹导入到工程中
3. 添加依赖库
 - Accelerate.framework
 - AudioToolbox.framework

初始化 SDK

1. 在需要使用地方导入头文件 GLPlayer.h
2. 加 SDK 初始化方法

```
[GLPlayerKit initWithSession: GLCoreKit.roomSession];
```

播放视频

1. 只有在房间 ID 与密码验证成功后，才能够正确播放视频

```
GLPlayerKit.delegate = self;  
[GLPlayerKit playWithView: 显示视频的 view];
```

2. 处理播放视频的回调

如果设置了 `delegate`，那么在调用完播放视频接口后，通常会收到以下回调

```
- (void)onPlayerStart:(NSString *)roomId
{
    //成功开始播放视频
}

- (void)onPlayerError:(NSString *)error roomId:(NSString *)roomId
{
    //尝试播放视频过程中出现了错误
}
```

3. 处理直播状态通知的回调

```
- (void)onLiveStart:(NSString *)roomId
{
    //直播开始的回调。如果GLPlayer的autoplay属性为YES，那么在收到这个回调
    的时候SDK底层会自动调用播放视频的接口尝试播放视频。如果为NO，那么需要开发者自
    行调用播放视频接口。
}

- (void)onLiveStop:(NSString *)roomId
{
    //直播结束的回调。
}
```

停止播放视频

需要停止视频播放时，调用

```
[GLPlayerKit stop];
```

打开/关闭视频声音

默认情况下播放视频时声音是会一起播放的，在播放过程中，如果需要关闭声音，可以调用

```
[GLPlayerKit setMute:YES];
```

恢复声音的话调用

```
[GLPlayerKit setMute:NO];
```

GotyeLivePublisher

集成步骤

1. 集成 GotyeLiveCore（如已集成略过这一步）。
2. 将 libGotyeLivePublisher.a 以及同级目录下的 include 文件夹导入到工程中
3. 添加依赖库
 - libic++.tbd（XCode6 及以下版本后缀名为 dylib）

初始化 SDK

1. 在需要使用地方导入头文件 GLPublisher.h
2. 添加 SDK 初始化方法

```
[GLPublisherKit initWithSession: GLCoreKit.roomSession];
```

登录主播账号

```
[GLPublisherKit loginWithForce:是否强制踢出当前登录账号
                  success:^( ) {
                      //登录成功回调
                  }
                  failure:^(NSError *error) {
                      //登录失败回调
                  }
];
```

开启预览

在开启直播之前，需要先打开摄像头，开启预览

```
[GLPublisherKit startPreview:视频渲染视图
                  success:^( ) {
                      //预览成功回调
                  }
                  failure:^(NSError *error) {
                      //预览失败回调
                  }
];
```

默认情况下，预览模式打开的是前置摄像头，需要指定开启的摄像头的话，可以用以下方式

```
[GLPublisherKit startPreview:视频渲染视图
                    withCameraState:指定的摄像头
                    success:^(()) {
                        //预览成功回调
                    }
                    failure:^(NSError *error) {
                        //预览失败回调
                    }
];
```

开始直播

预览开启后，就可以正式开始直播了

```
[GLPublisherKit publish];
```

默认情况下，视频采样的分辨率为 640*480，如果需要更改为别的分辨率，可以设置 videoSize

```
[GLPublisherKit setVideoSize:CGSizeMake(320, 568)];
```

需要注意的是，videoSize 的设置需要在下一次开启直播才会生效。

直播过程中如果需要切换摄像头

```
[GLPublisherKit toggleCamera];
```

或者切换麦克风状态

```
[GLPublisherKit setMute:YES];
```

都是可以实时生效的。

结束直播

如果只是单纯的结束视频直播，可以只调用

```
[GLPublisherKit unpublish];
```

之后会回到视频预览状态，并且主播账号也不会登出。

如果需要完全退出的话，调用

```
[GLPublisherKit stop];
```