
目錄

概述	1.1
如何获取access_token	1.2
SDK集成	1.3
开发环境配置	1.3.1
大班直播	1.3.2
大班互动	1.3.3
小班直播	1.3.4
生活直播	1.3.5
点播	1.3.6
离线下载	1.3.7
API说明	1.4
公共API	1.4.1
直播API	1.4.2
房间	1.4.2.1
视频	1.4.2.2
画板	1.4.2.3
直播状态	1.4.2.4
成员	1.4.2.5
聊天	1.4.2.6
提问	1.4.2.7
投票	1.4.2.8
抽奖	1.4.2.9
鲜花	1.4.2.10
公告	1.4.2.11
广播	1.4.2.12
网络线路	1.4.2.13
音视频互动	1.4.2.14
小班API	1.4.2.15
连麦	1.4.2.15.1
生活直播API	1.4.2.16
点赞	1.4.2.16.1
商城	1.4.2.16.2
商品	1.4.2.16.3
邀请榜单	1.4.2.16.4

举报	1.4.2.16.5
生活直播模型说明	1.4.2.16.6
点播API	1.4.3
房间	1.4.3.1
聊天	1.4.3.2
视频	1.4.3.3
画板	1.4.3.4
问答	1.4.3.5
点播线路切换	1.4.3.6
片头广告	1.4.3.7
广播	1.4.3.8
画板	1.4.3.9
音视频互动	1.4.3.10
常见问题	2.1
本地网络隐私	2.1.1
SDK不再支持处理器是32位的设备	2.1.2
关于项目不能使用CocoaPods	2.1.3
H5视频默认全屏的处理	2.1.4
退出直播间重新进入崩溃	2.1.5
横竖屏设置	2.1.6
playbackID的获取及用途	2.1.7
Apple Mach-O Linker Error	2.1.8
Failed to create provisioning profile	2.1.9
App Transport Security	2.1.10
Bitcode	2.1.11
回放聊天如何自动跟随	2.1.12
如何支持后台播放	2.1.13
如何获取播放缓冲情况	2.1.14
关于ReactNative、APICloud或其他非原生语言的接入	2.1.15
SDK错误捕捉	2.1.16
SDK下载	2.2
SDK文档下载	2.3

简介

欢拓云课堂iOS SDK 是一套基于armv7s、arm64 处理器设备的应用程序接口，提供直播点播功能，不仅支持默认官方UI，并且支持自定义UI。您可以使用欢拓云课堂iOS SDK 让您的APP直接拥有观看直播、观看回放的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

前置条件

- 已经与欢拓网络公司达成协议
- 有获取access_token的途径
 - 获取access_token参数，详情参考:[如何与我的APP进行对接](#)

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

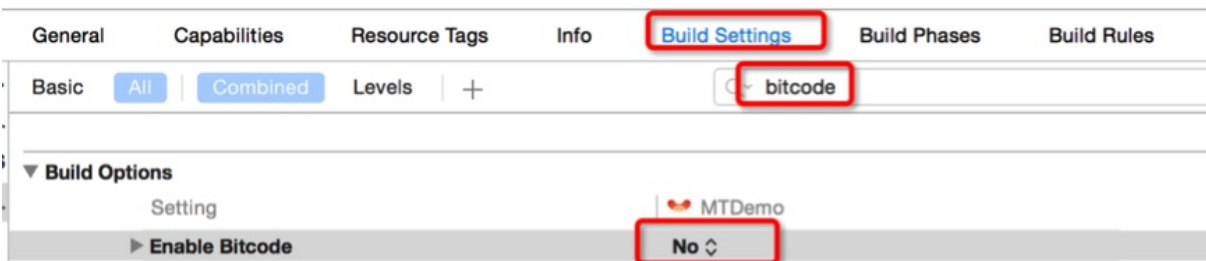
快速开始

开发环境配置

- 支持https
 - 在Info.plist中添加NSAppTransportSecurity类型Dictionary。在NSAppTransportSecurity下添加NSAllowsArbitraryLoads类型Boolean, 值为YES



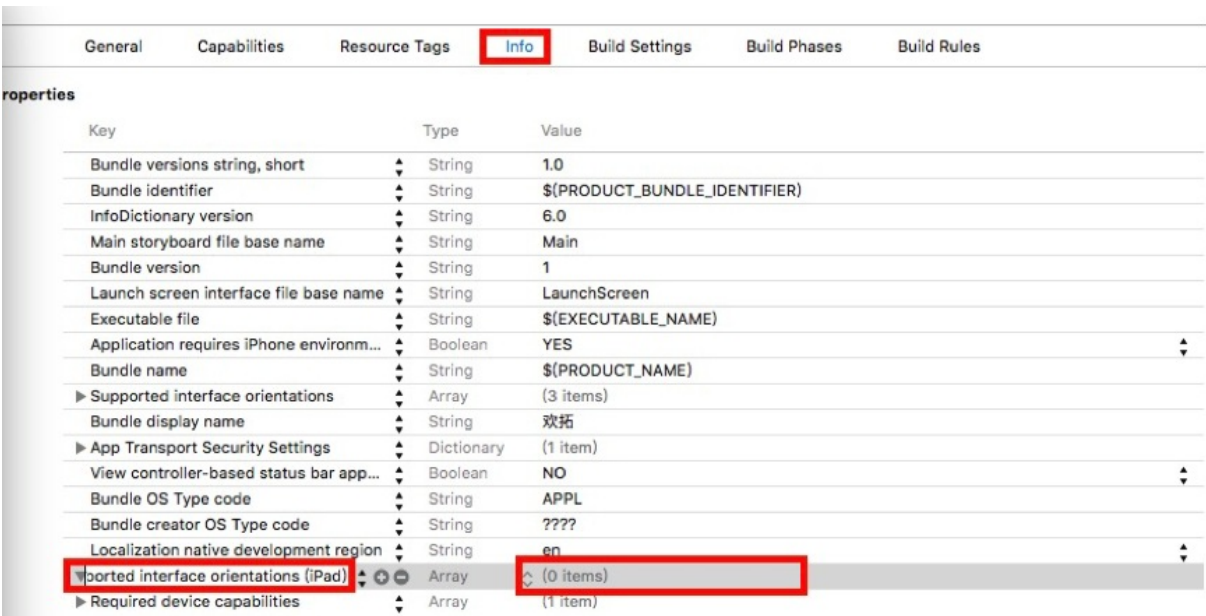
- 关闭bitcode
 - 在Xcode项目里面的 Build Settings -> Build Options -> Enable Bitcode , 设置为 No



- status Bar设置可以隐藏 在info.plist文件里面添加 View controller-based status bar appearance 把值设置为 NO



- ipad的兼容 要把Supported interface orientations (iPad)里面的东西都删除



- CocoaPods 方式集成SDK
- platform :ios, '9.0'
- target '你的项目名' do

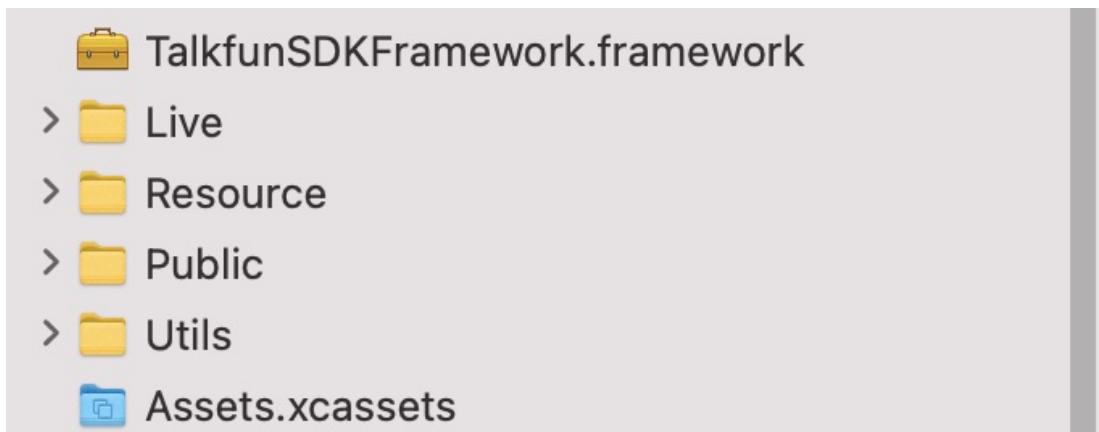
- use_frameworks! #欢拓SDK的依赖
- pod 'TalkfunSDKFramework', '~> 4.4.9' #欢拓SDK
- 使用欢拓云课堂SDK，必须传入一个access_token，该access_token的获取，参考[申请密钥流程](#)
- 如需后台播放，请参考[后台播放配置](#)

到此，你已完成了SDK的依赖添加

Copyright Talkfun all right reserved，powered by Gitbook修订时间：2021-11-17 10:40:21

大班直播接入

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步
 - 把 **TalkfunSDKFramework.framework** 整个文件夹拖入工程
 - 把 **Live** 整个文件夹拖入工程
 - 把 **Resource** 整个文件夹拖入工程
 - 把 **Public**（工具类）整个文件夹拖入工程
 - 把 **Utils**（工具类）整个文件夹拖入工程
 - 把 **Assets.xcassets**文件夹内的图片拖入工程



初始化SDK逻辑

代码演示:

- 特别说明的是，在实例化TalkfunSDKLive对象时首先要有access_token
- 需要提供一个摄像头容器和一个ppt容器

```
#import "TalkfunViewController.h"
#import <TalkfunSDKFramework/TalkfunSDKFramework.h>

@interface TalkfunViewController ()<TalkfunSDKLiveDelegate,TalkfunSDKRtcDelegate>

@property (nonatomic, strong) TalkfunSDKLive * talkfunSDK;
//摄像头容器
@property (nonatomic, strong) UIView * cameraView;
//ppt容器
@property (nonatomic, strong) UIView * pptView;

@end
```



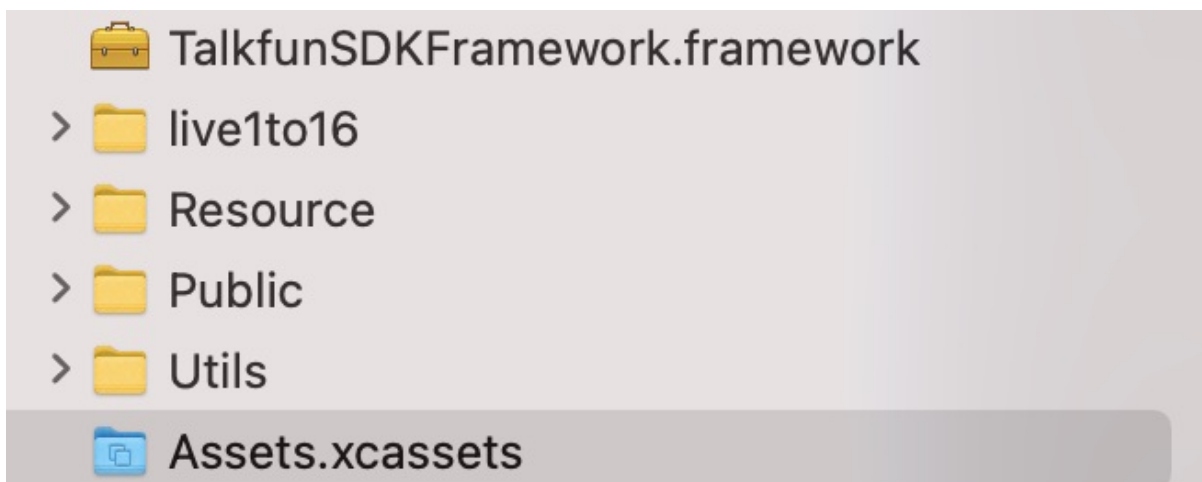
```
//TODO:初始化SDK-----  
self.talkfunSDK = [TalkfunSDKLive sharedInstance];  
//配置初始化参数  
TalkfunSDKLiveInitParams *params = [[TalkfunSDKLiveInitParams alloc] init];  
params.accessToken = token;  
params.courseType = TalkfunCourseTypeLive;  
//进入房间  
[self.talkfunSDK enterRoom:params];  
  
//配置摄像头  
[self.talkfunSDK configureCameraContainerView:self.cameraView];  
//配置画板  
[self.talkfunSDK configurePPTContainerView:self.pptView];  
//设置代理  
self.talkfunSDK.delegate = self;
```

到此为止已经简单地实例化了SDK，已经具有播放音视频以及显示ppt的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

大班互动接入

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步
 - 把 **TalkfunSDKFramework.framework** 整个文件夹拖入工程
 - 把 **live1to16** 整个文件夹拖入工程
 - 把 **Resource** 整个文件夹拖入工程
 - 把 **Public**（工具类）整个文件夹拖入工程
 - 把 **Utils**（工具类）整个文件夹拖入工程
 - 把 **Assets.xcassets**文件夹内的图片拖入工程



初始化SDK逻辑

代码演示:

- 特别说明的是，在实例化TalkfunSDKLive对象时首先要有access_token
- 需要提供一个摄像头容器和一个ppt容器

```
#import "TalkfunInteractLive.h"
#import <TalkfunSDKFramework/TalkfunSDKFramework.h>

@interface TalkfunInteractLive ()<TalkfunSDKLiveDelegate, TalkfunSDKRtcDelegate>

@property (nonatomic, strong) TalkfunSDKLive * talkfunSDK;
//摄像头容器
@property (nonatomic, strong) UIView * cameraView;
//ppt容器
@property (nonatomic, strong) UIView * pptView;

@end
```

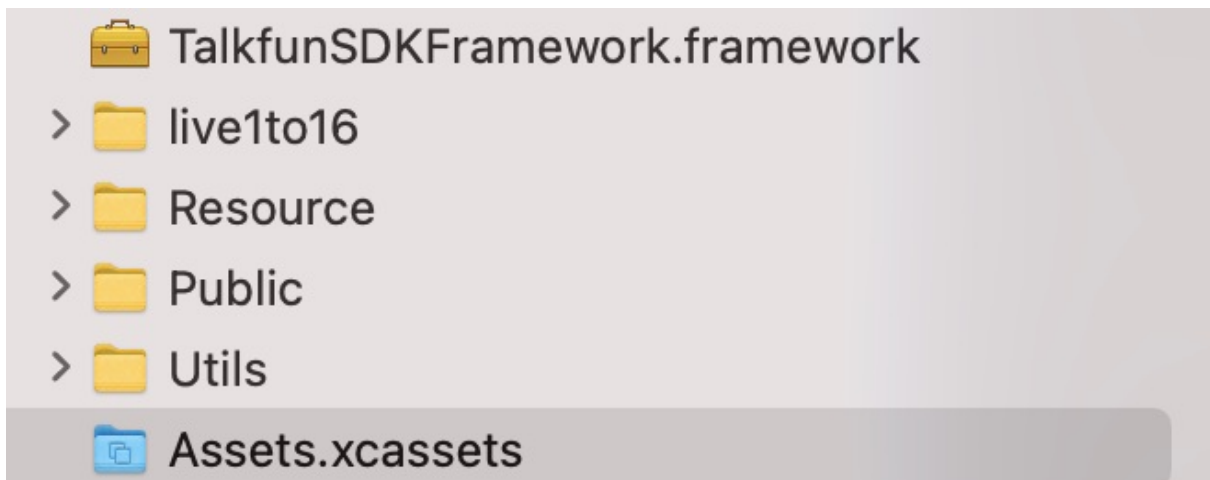
```
//TODO:初始化SDK-----  
self.talkfunSDK = [TalkfunSDKLive sharedInstance];  
//配置初始化参数  
TalkfunSDKLiveInitParams *params = [[TalkfunSDKLiveInitParams alloc]init];  
params.accessToken = token;  
params.courseType = TalkfunCourseTypeInteractiveLive;  
//进入房间  
[self.talkfunSDK enterRoom:params];  
  
//配置摄像头  
[self.talkfunSDK configureCameraContainerView:self.cameraView];  
//配置画板  
[self.talkfunSDK configurePPTContainerView:self.pptView];  
//设置代理  
self.talkfunSDK.delegate = self;  
//小班的代理  
self.talkfunSDK.rtcManager.delegate = self;
```

到此为止已经简单地实例化了SDK，已经具有播放音视频以及显示ppt的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

小班互动接入

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步
 - 把 **TalkfunSDKFramework.framework** 整个文件夹拖入工程
 - 把 **live1to16** 整个文件夹拖入工程
 - 把 **Resource** 整个文件夹拖入工程
 - 把 **Public**（工具类）整个文件夹拖入工程
 - 把 **Utils**（工具类）整个文件夹拖入工程
 - 把 **Assets.xcassets**文件夹内的图片拖入工程



初始化SDK逻辑

代码演示:

- 特别说明的是，在实例化TalkfunSDKLive对象时首先要有access_token
- 需要提供一个摄像头容器和一个ppt容器

```
#import "Talkfun1to16Live.h"
#import <TalkfunSDKFramework/TalkfunSDKFramework.h>

@interface Talkfun1to16Live ()<TalkfunSDKLiveDelegate,TalkfunSDKRtcDelegate>

@property (nonatomic,strong) TalkfunSDKLive * talkfunSDK;
//摄像头容器
@property (nonatomic,strong) UIView * cameraView;
//ppt容器
@property (nonatomic,strong) UIView * pptView;

@end
```

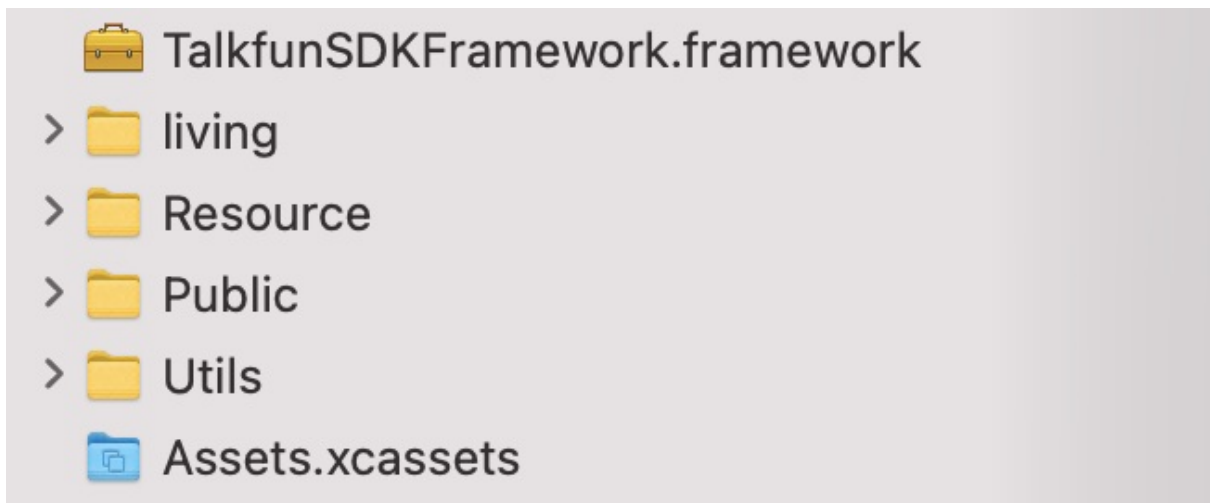
```
//TODO:初始化SDK-----  
self.talkfunSDK = [TalkfunSDKLive sharedInstance];  
//配置初始化参数  
TalkfunSDKLiveInitParams *params = [[TalkfunSDKLiveInitParams alloc] init];  
params.accessToken = token;  
params.courseType = TalkfunCourseTypeStatusRTC;  
//进入房间  
[self.talkfunSDK enterRoom:params];  
  
//配置摄像头  
[self.talkfunSDK configureCameraContainerView:self.cameraView];  
//配置画板  
[self.talkfunSDK configurePPTContainerView:self.pptView];  
//设置代理  
self.talkfunSDK.delegate = self;  
//小班的代理  
self.talkfunSDK.rtcManager.delegate = self;
```

到此为止已经简单地实例化了SDK，已经具有播放音视频以及显示ppt的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

生活直播接入

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步
 - 把 **TalkfunSDKFramework.framework** 整个文件夹拖入工程
 - 把 **living** 整个文件夹拖入工程
 - 把 **Resource** 整个文件夹拖入工程
 - 把 **Public**（工具类）整个文件夹拖入工程
 - 把 **Utils**（工具类）整个文件夹拖入工程
 - 把 **Assets.xcassets**文件夹内的图片拖入工程



初始化SDK逻辑

代码演示:

- 特别说明的是，在实例化TalkfunSDKLive对象时首先要有access_token
- 需要提供一个摄像头容器和一个ppt容器

```
#import "TalkfunLivingController.h"
#import <TalkfunSDKFramework/TalkfunSDKFramework.h>

@interface TalkfunLivingController ()<TalkfunSDKLifeLiveDelegate>

@property (nonatomic, strong) TalkfunSDKLive * talkfunSDK;
//摄像头容器
@property (nonatomic, strong) UIView * cameraView;
//ppt容器
@property (nonatomic, strong) UIView * pptView;

@end
```

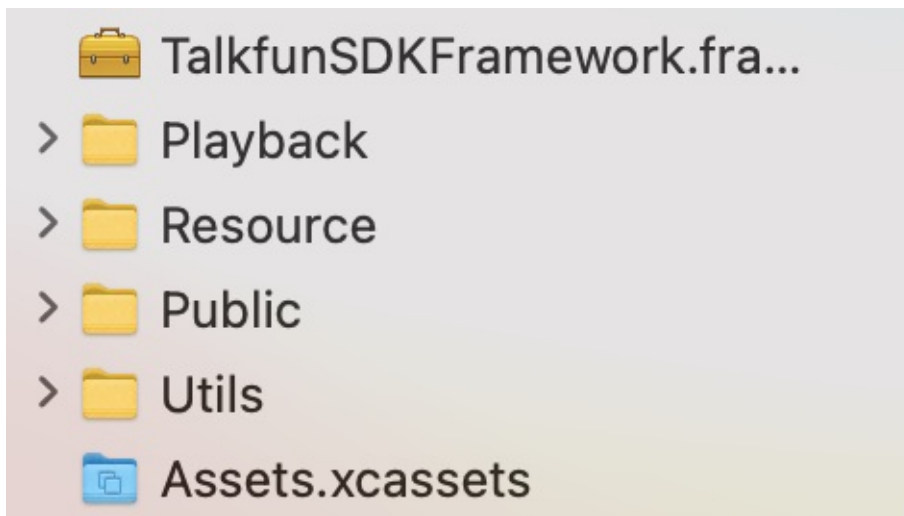
```
//TODO:初始化SDK-----  
self.talkfunSDK = [TalkfunSDKLive sharedInstance];  
//配置初始化参数  
TFInitParams *params = [[TFInitParams alloc]init];  
params.accessToken = token;  
//进入房间  
[self.talkfunSDK enterRoom:params];  
  
//配置摄像头  
self.livingSDK.cameraContainerView = self.cameraView;  
//配置画板  
[self.talkfunSDK configurePPTContainerView:self.pptView];  
//设置代理  
self.talkfunSDK.delegate = self;
```

到此为止已经简单地实例化了SDK，已经具有播放音视频以及显示ppt的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

点播接入

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步
 - 把 **TalkfunSDKFramework.framework** 整个文件夹拖入工程
 - 把 **Playback** 整个文件夹拖入工程
 - 把 **Resource** 整个文件夹拖入工程
 - 把 **Public**（工具类）整个文件夹拖入工程
 - 把 **Utils**（工具类）整个文件夹拖入工程
 - 把 **Assets.xcassets**文件夹内的图片拖入工程



初始化SDK逻辑

代码演示:

- 特别说明的是，在实例化TalkfunSDKLive对象时首先要有access_token
- 需要提供一个摄像头容器和一个ppt容器

```
#import "TalkfunPlaybackViewController"  
#import <TalkfunSDKFramework/TalkfunSDKFramework.h>  
  
@interface TalkfunPlaybackViewController ()<TalkfunSDKPlaybackDelegate>  
  
@property (nonatomic, strong) TalkfunSDKPlayback * talkfunSDK;  
//摄像头容器  
@property (nonatomic, strong) UIView * cameraView;  
//ppt容器  
@property (nonatomic, strong) UIView * pptView;  
  
@end
```



```
//TODO:初始化SDK-----  
self.talkfunSDK = [TalkfunSDKPlayback sharedInstance];  
//配置初始化参数  
TalkfunSDKPlaybackInitParams *params = [[TalkfunSDKPlaybackInitParams alloc]init];  
params.accessToken = token;  
  
//进入房间  
[self.talkfunSDK enterRoom:params];  
  
//配置摄像头  
[self.talkfunSDK configureCameraContainerView:self.cameraView];  
//配置画板  
[self.talkfunSDK configurePPTContainerView:self.pptView];  
//设置代理  
self.talkfunSDK.delegate = self;
```

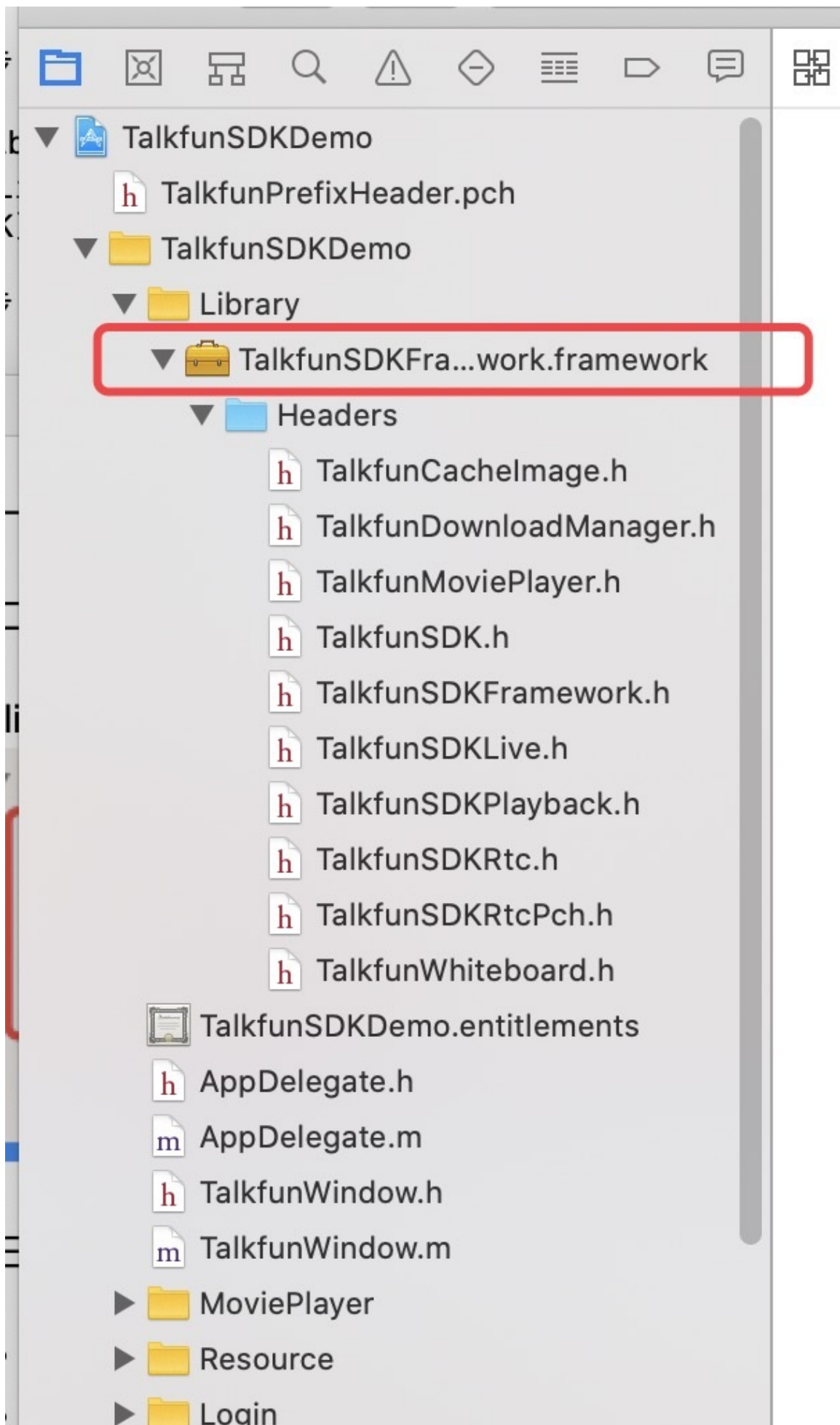
到此为止已经简单地实例化了SDK，已经具有播放音视频以及显示ppt的功能。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:57

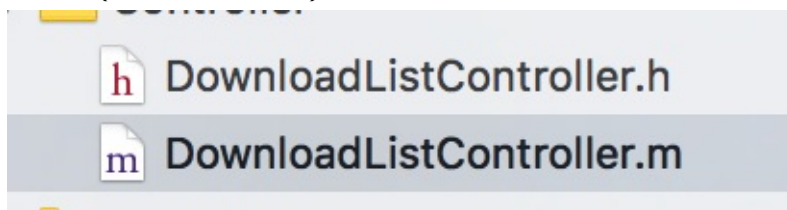
离线下载实现逻辑

- 第一步是先按照[开发环境配置](#)搭建好工程环境
- 第二步 (SDK)

把 TalkfunSDKFramework.framework 文件拖入工程。



- 第三步（点播下载控制器）View Controller 如图所示：



添加引用

- 在上述DownloadListController.m中添加引用#import "TalkfunDownloadManager.h"

添加声明属性

- 在DownloadListController.m中添加属性
 - TalkfunDownloadManager * downloadManager

```
#import "DownloadListController.h"
#import "TalkfunDownloadManager.h"

@interface DownloadListController ()

//下载管理对象
@property (nonatomic, strong) TalkfunDownloadManager * downloadManager;

@end
```

创建下载的对象

- 以单例的形式实例化下载对象

```
self.downloadManager = [TalkfunDownloadManager shareManager];
```

创建下载任务并启动下载

- 支持将课程数据下载到本地，在无网络环境正常播放。
- 进入回放时，需要传入回放的access_token和playbackID，SDK会根据传入的access_token和playbackID从本地检索是否有该课程的缓存文件，如果有缓存则读取本地缓存播放，否则去请求网络。

```
//创建下载任务
[self.downloadManager appendDownloadWithAccessToken: access_token playbackID:self.playbackID title:nil];

//启动下载
[self.downloadManager startDownload:self.playbackID];
```

下载进度

- TalkfunDownloadManagerDelegate

```
- (void)downloadProgress:(TalkfunDownloadProgressModel*_Nullable)progressModel
```

- TalkfunDownloadProgressModel 参数

```
@interface TalkfunDownloadProgressModel : NSObject
@property(nonatomic, strong)NSString*_Nullable playbackID;//课程id
@property(nonatomic, assign)CGFloat      totalSize;//课程总大小
@property(nonatomic, assign)CGFloat      downloadedSize;//已下载的大小
@end
```

下载状态

- TalkfunDownloadManagerDelegate

```
- (void)downloadStatus:(TalkfunDownloadStatusModel*_Nullable)statusModel
```

- TalkfunDownloadStatusModel 参数

```
@interface TalkfunDownloadStatusModel : NSObject
@property(nonatomic, strong)NSString*_Nullable playbackID;//课程id
@property(nonatomic, assign)CGFloat      downloadedSize;//已下载的大小
@property(nonatomic, assign)CGFloat      duration;          //课程长度
@property(nonatomic, assign)CGFloat      totalSize;          ///课程总大小
@property(nonatomic, assign)TalkfunDownloadStatus downloadStatus;
@property (nonatomic, copy) NSString *_Nullable thumb;
@end
```

下载发生错误

- TalkfunDownloadManagerDelegate

```
- (void)downloadError:(TalkfuError *_Nullable)error
```

- TalkfuError 参数

```
@interface TalkfuError : NSObject
@property(nonatomic, strong)NSString *_Nullable playbackID;//课程id
@property(nonatomic, strong)NSString *_Nullable reasonDescription;//原因描述
@property(nonatomic, assign)NSString *_Nullable statusCode;//状态码
@property(nonatomic, assign)NSString *_Nullable URL;
@end
```

下载完成后可通过课程access_token离线观看

实例:

```
NSArray * playbackList = self.downloadManager.getDownloadList;

TalkfunDownloadListModel * model = playbackList[indexPath.row];

if (model.downloadStatus == TalkfunDownloadStatusFinish) {

    //TODO:设置点播控制器数据
    TalkfunPlaybackViewController * playbackVC = [TalkfunPlaybackViewController new];
    playbackVC.modalPresentationStyle = UIModalPresentationFullScreen;
    playbackVC.playbackID = model.playbackID;
    playbackVC.res = @{@"data":@{"access_token":model.access_token},TalkfunPlaybackID:model
.playbackID};
    playbackVC.downloadCompleted = YES;//下载完成
    [self presentViewController:playbackVC animated:NO completion:nil];

}
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 18:38:37

功能使用

如果你有其它功能的需求，可以查询下列功能列表或查看[TalkfunSDK.h](#)文件，下载功能请查看[TalkfunDownloadManager.h](#)

播放状态获取及回调

- @property (nonatomic,assign,readonly) TalkfunPlayStatus playStatus;

在TalkfunSDK中，通过talkfunSDK的状态来反馈播放状态。我们定义了集中状态，确保talkfunSDK对象在有限的几个状态间切换。

状态名	含义
TalkfunPlayStatusStop	初始化时或播放完毕时指定的状态
TalkfunPlayStatusPlaying	正在播放中的状态
TalkfunPlayStatusPause	播放暂停
TalkfunPlayStatusSeeking	seek的状态
TalkfunPlayStatusError	播放错误

playStatus状态对应的delegate回调方法

```
- (void)talkfunPlayStatusChange:(TalkfunPlayStatus)status;
```

监听事件（聊天、提问、公告、广播等）

```
- (void)on:(NSString *)event withCallback:(void (^)(id obj))callback;
```

示例代码：

```
//发送聊天
[self.talkfunSDK on:@"chat:send" withCallback:^(id obj) {
    [[NSNotificationCenter defaultCenter] postNotificationName:@"chat" object:nil
        userInfo:@{@"mess":obj}];
}];
```

接口及参数说明：

[直播接口及参数](#)

[点播接口及参数](#)

发送信息（聊天、提问、送花、投票、获取投票数据）

```
- (void)emit:(NSString *)event withParameter:(NSDictionary *)parameter withCallback:(void
    (^)(id obj))callback;
```

示例代码：

```
[self.talkfunSDK emit:@"question:ask" withParameter:@{@"msg":self.askTF.text}
withCallback:^(id obj) {

    //发送完成后的回调

}];
```

接口及参数说明：

[接口及参数](#)

网络选择

- 如果当前网络速度不理想，SDK提供了接口更改网络。

```
//获取网络列表
- (void)getNetworkList;

//设置网络
- (void)setNetwork:(NSString *)operatorID;
```

调用示例：

- 首先用on:withCallback:方法监听network:select事件


```
//网路选择
[self.talkfunSDK on:@"network:select" withCallback:^(id obj) {

    //obj:
    {
        code = 0;
        data = (
            {
                key = auto;
                name = "\U81ea\U52a8\U9009\U62e9\U6700\U4f73\U7ebf\U8def";
            },
            {
                key = telecom;
                name = "\U4e2d\U56fd\U7535\U4fe1";
            },
            {
                key = netcom;
                name = "\U4e2d\U56fd\U8054\U901a";
            },
            {
                key = mobile;
                name = "\U4e2d\U56fd\U79fb\U52a8";
            },
            {
                key = teletron;
                name = "\U7535\U4fe1\U901a";
            },
            {
                key = greatwall;
                name = "\U957f\U57ce\U5bbd\U5e26";
            },
            {
                key = education;
                name = "\U6559\U80b2\U7f51";
            },
            {
                key = cttnet;
                name = "\U4e2d\U56fd\U94c1\U901a";
            }
        );
        network = {
            ip = "219.136.204.101";
            isp = "\U7535\U4fe1";
            location = "\U4e2d\U56fd\U5e7f\U4e1c\U5e7f\U5dde";
        };
    }
}];
```

参数	类型	描述
code	NSNumber	状态码
data	NSArray	数据
key	NSString	运营商的key
Name	NSString	运营商的名字

- 其次调用- (void)getNetworkList;接口，调用完SDK就会把数据回调回来，就在第一步监听的network:select事件里面。

```
[self.talkfunSDK getNetworkList];
```

- 最后调用- (void)setNetwork:(NSString *)operatorID;接口，把对应的运营商的key作为参数传进去。

```
[self.talkfunSDK setNetwork:key];
```

点播倍数播放

- SDK新增倍数播放功能，传入的数值为0~2之间的数值即可，传入0即为暂停。

```
//点播播放速度 [0-2]
@property (nonatomic) float playbackRate;
```

点播线路切换

```
/**
 * 获取网络列表(点播)
 *
 */
- (NSArray *)getNetworkLinesList;

/**
 * 设置播放对应的线路(点播)
 *
 * 把相应线路传进SDK
 */
- (void)setNetworkLine:(NSNumber *)networkLineIndex;
```

参数	类型	描述
current	NSNumber	1为当前线路
index	NSNumber	线路相应指针（作为设置线路参数）
name	NSString	线路的名字

示例：

```
//获取线路列表
NSArray * list = [self.talkfunSDK getNetworkLinesList];
//拿到相应线路指针
NSNumber * index = list.firstObject[@"index"];
//设置线路
[self.talkfunSDK setNetworkLine:index];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

公共定义说明

TalkfunSDK.h

```
extern NSString * const TalkfunSDKVersion;

/**
 * 用户事件 (public)
 */
extern NSString * const TALKFUN_EVENT_PLAY;           //播放
extern NSString * const TALKFUN_EVENT_PAUSE;         //暂停
extern NSString * const TALKFUN_EVENT_STOP;          //停止
extern NSString * const TALKFUN_EVENT_CAMERA_SHOW;   //摄像头开
extern NSString * const TALKFUN_EVENT_CAMERA_HIDE;  //摄像头关
extern NSString * const TALKFUN_EVENT_DESKTOP_START; //桌面分享开始
extern NSString * const TALKFUN_EVENT_DESKTOP_STOP; //桌面分享结束
extern NSString * const TALKFUN_EVENT_DESKTOP_PAUSE; //桌面分享暂停
extern NSString * const TALKFUN_EVENT_MODE_CHANGE;  //直播间模式切换
extern NSString * const TALKFUN_EVENT_NETWORK_SPEED; //网络速度

/**
 * 用户事件 (live)
 */
extern NSString * const TALKFUN_EVENT_VOTE_NEW;      //投票开始
extern NSString * const TALKFUN_EVENT_VOTE_PUB;     //投票结束
extern NSString * const TALKFUN_EVENT_VOTE_DEL;     //投票删除
extern NSString * const TALKFUN_EVENT_VOTE_INFO;    //查看投票信息
extern NSString * const TALKFUN_EVENT_VOTE_SUBMIT;  //提交投票
extern NSString * const TALKFUN_EVENT_LOTTERY_START; //抽奖开始
extern NSString * const TALKFUN_EVENT_LOTTERY_STOP; //抽奖结束

extern NSString * const TALKFUN_SIGN_NEW;           //签到开始
extern NSString * const TALKFUN_SIGN_END;           //签到结束
extern NSString * const TALKFUN_SIGN_IN;           //提交签到

extern NSString * const TALKFUN_EVENT_BROADCAST;    //公共广播
extern NSString * const TALKFUN_EVENT_CHAT_DISABLE; //禁言
extern NSString * const TALKFUN_EVENT_CHAT_ENABLE; //解除禁言
extern NSString * const TALKFUN_EVENT_CHAT_DISABLE_ALL; //全体禁言
extern NSString * const TALKFUN_EVENT_CHAT_SEND;    //信息发送接收
extern NSString * const TALKFUN_EVENT_CHAT_PRIVATE; //私聊
extern NSString * const TALKFUN_EVENT_FLOWER_SEND;  //送花
extern NSString * const TALKFUN_EVENT_FLOWER_INIT;  //送花初始化
extern NSString * const TALKFUN_EVENT_FLOWER_TOTAL; //花朵总数
extern NSString * const TALKFUN_EVENT_FLOWER_TIME_LEFT; //送花剩余时间
extern NSString * const TALKFUN_EVENT_QUESTION_LIST; //问题列表
extern NSString * const TALKFUN_EVENT_QUESTION_ASK; //提问
extern NSString * const TALKFUN_EVENT_QUESTION_AUDIT; //通过
extern NSString * const TALKFUN_EVENT_QUESTION_REPLY; //问题回复
```

```

extern NSString * const TALKFUN_EVENT_QUESTION_DELETE; //问题删除
extern NSString * const TALKFUN_EVENT_MEMBER_JOIN_ME; //我进入房间
extern NSString * const TALKFUN_EVENT_MEMBER_JOIN_OTHER; //其他人进入房间
extern NSString * const TALKFUN_EVENT_MEMBER_LEAVE; //其他人离开房间
extern NSString * const TALKFUN_EVENT_MEMBER_TOTAL; //房间总人数
extern NSString * const TALKFUN_EVENT_MEMBER_KICK; //踢人
extern NSString * const TALKFUN_EVENT_MEMBER_FORCEOUT; //被强迫退出房间
extern NSString * const TALKFUN_EVENT_MEMBER_LIST; //用户列表
extern NSString * const TALKFUN_EVENT_ANNOUNCE_NOTICE; //公告
extern NSString * const TALKFUN_EVENT_ANNOUNCE_ROLL; //滚动公告
extern NSString * const TALKFUN_EVENT_LIVE_WAIT; //直播未开始
extern NSString * const TALKFUN_EVENT_LIVE_START; //直播开始
extern NSString * const TALKFUN_EVENT_LIVE_STOP; //直播结束
extern NSString * const TALKFUN_EVENT_ROOM_INIT; //房间初始化

extern NSString * const TALKFUN_EVENT_RTC_INVITE; //被邀请上讲台
extern NSString * const TALKFUN_EVENT_RTC_INVITECANCEL; //取消邀请上讲台

extern NSString * const TALKFUN_EVENT_CURRENT_VIDEO_DEFINITION_LIST; //默认视频清晰度
/**
 * 用户事件 ( playback )
 */
extern NSString * const TALKFUN_EVENT_VOD_START; //点播开始
extern NSString * const TALKFUN_EVENT_VOD_STOP; //点播结束
extern NSString * const TALKFUN_EVENT_VOD_DURATION; //点播总时长
extern NSString * const TALKFUN_EVENT_VOD_INFO; //点播信息
extern NSString * const TALKFUN_EVENT_VOD_QUESTION_APPEND; //追加问题列表
extern NSString * const TALKFUN_EVENT_VOD_QUESTION_REFRESH; //刷新问题列表
extern NSString * const TALKFUN_EVENT_VOD_QUESTION_SCROLL; //获得问答滚动的索引, 必须使用TALKFUN_EVENT_VOD_QUESTION_REFRESH获得的数据列表
extern NSString * const TALKFUN_EVENT_VOD_MESSAGE_APPEND; //聊天信息列表
extern NSString * const TALKFUN_EVENT_VOD_MESSAGE_REFRESH; //刷新聊天列表
extern NSString * const TALKFUN_EVENT_VOD_MESSAGE_SCROLL; //获得聊天滚动的索引, 必须使用TALKFUN_EVENT_VOD_MESSAGE_REFRESH获得的数据列表
extern NSString * const TALKFUN_EVENT_VOD_CHAPTER_LIST; //章节列表

extern NSString * const TALKFUN_EVENT_WHITEBOARD_PAGE_FRAME; //画板当前页的白板与ppt的frame

extern NSString * const TALKFUN_EVENT_SCORE_CONFIG; //评分配置
extern NSString * const TALKFUN_EVENT_PPT_DISPLAY; //ppt是否显示

/**
 * 用户事件 ( download )
 */
extern NSString * const TALKFUN_EVENT_DOWNLOAD_STATUS; //下载状态
extern NSString * const TALKFUN_EVENT_DOWNLOAD_PROGRESS; //下载进度

//extern NSString * const TALKFUN_EVENT_INSUFFICIENT_DISK_SPACE_WARNING; //警告磁盘空间不足

```

```

/**
 * 广播名
 */
extern NSString * const TalkfunErrorNotification;           //错误信息广播 (类型看错误信息类型)

/**
 用户角色
 */
extern NSString * const TalkfunMemberRoleSpadmin;         //超级管理员(老师)
extern NSString * const TalkfunMemberRoleAdmin;          //管理员(助教)
extern NSString * const TalkfunMemberRoleUser;           //普通用户(学生)
extern NSString * const TalkfunMemberRoleGuest;          //游客

extern int const TalkfunMemberGenderUnknow;              //未知性别
extern int const TalkfunMemberGenderMale;                //男性
extern int const TalkfunMemberGenderFemale;              //女性

extern NSString * const TalkfunPlaybackID;                //点播离线播放的key

@interface TalkfunSDK : NSObject

#ifdef TalkfunError

//错误信息类型
typedef enum {
    //详细错误代码
    TalkfunErrorCourseIsFinish = 1262,                    //课程已结束
    TalkfunErrorAuthenticationFail = 10010,               //鉴权失败,access_token错误或者过期

    //错误大类
    TalkfunErrorLoadingData = 200000,                     //加载数据出错
    TalkfunErrorNetworkDisconnect = 200001,               //网络连接错误
    TalkfunErrorVideoError = 200002,                     //视频加载错误
    TalkfunErrorDownloadError = 200003,                   //下载错误
    TalkfunErrorLoadingOfflineFileError = 200004,         //加载离线文件错误
    TalkfunErrorConnectionFailedError = 200005,          //socket连接失败
    TalkfunErrorNoMoreSpace = 10035,                       //房间已满
    TalkfunErrorKick = 10034 //你已被管理员移出房间

}TalkfunError;

#endif

//视频加载状态
typedef NS_OPTIONS(NSUInteger, TalkfunPlayerLoadState) {
    TalkfunPlayerLoadStateUnknown = 0,

```

```
    TalkfunPlayerLoadStatePlayable      = 1 << 0,
    TalkfunPlayerLoadStatePlaythroughOK = 1 << 1, // Playback will be automatically started in this state when shouldAutoplay is YES
    TalkfunPlayerLoadStateStalled      = 1 << 2, // Playback will be automatically paused in this state, if started
};

//视频播放结束原因
typedef NS_ENUM(NSUInteger, TalkfunPlayerMovieFinishReason) {
    TalkfunPlayerMovieFinishReasonPlaybackEnded,
    TalkfunPlayerMovieFinishReasonPlaybackError,
    TalkfunPlayerMovieFinishReasonUserExited
};

//状态码
typedef enum{

    TalkfunCodeSuccess = 0,
    TalkfunCodeFailed  = -1,
    TalkfunCodeTimeout = -100,
    TalkfunCodeGetDataFail = -101,
    TalkfunCodeWrongPassword = -3,
    TalkfunCodeUnknownError = 1000

}TalkfunCode;

//播放状态
typedef enum {

    TalkfunPlayStatusStop      = 0,
    TalkfunPlayStatusPlaying   = 1,
    TalkfunPlayStatusPause     = 2,
    TalkfunPlayStatusSeeking   = 3,
    TalkfunPlayStatusError     = 4

}TalkfunPlayStatus;

//0 : 应用 , 1 : 播放 , 2 : 暂停 , 3 : 拖动 , 4 : 关闭
//多媒体播放状态(音视频)
typedef enum {

    TalkfunMultiMediaStatusChangeListenerPlaying      = 1, //播放
    TalkfunMultiMediaStatusChangeListenerStop         = 2, //停止
    TalkfunMultiMediaStatusChangeListenerSeek         = 3, //快进
    TalkfunMultiMediaStatusChangeListenerClose        = 4, //关闭
    TalkfunMultiMediaStatusChangeListenerComplate     = 5, //完成
    TalkfunMultiMediaStatusChangeListenerError        = 6, //错误

}TalkfunMultiMediaStatusChangeListener;

//视频模式
typedef enum {
```

```
    TalkfunLiveModeVideo    = 0,
    TalkfunLiveModeDesktop = 2,
    TalkfunLiveModeRTC= 3
}TalkfunLiveMode;

//上课模式
typedef enum {

    TalkfunPageModeCourse    = 0, //课件
    TalkfunPageModeKeyNote = 1, //主讲
    TalkfunPageModeGallery= 2 //画廊模式
}TalkfunPageMode;

//网络状态
typedef enum {

    TalkfunNetworkStatusGood    = 0, //网络好
    TalkfunNetworkStatusGeneral = 1, //网络一般
    TalkfunNetworkStatusBad     = 2 //网络较差

}TalkfunNetworkStatus;

//日志输出控制
typedef enum {
    TalkfunLoggerLevelAll = 0,
    TalkfunLoggerLevelLog = 1,
    TalkfunLoggerLevelInfo = 2,
    TalkfunLoggerLevelDebug = 3,
    TalkfunLoggerLevelWarn = 4,
    TalkfunLoggerLevelError = 5,
    TalkfunLoggerLevelFatal = 6,
    TalkfunLoggerLevelNone = 10
}TalkfunLoggerLevel;

typedef enum {
    TalkfunLiveStatusWait    = 0, //直播未开始
    TalkfunLiveStatusStart  = 1, //直播开始
    TalkfunLiveStatusStop    = 2, //直播结束

}TalkfunLiveStatus;

typedef enum {
    TalkfunCourseTypeUnknown = -1, //未知
    TalkfunCourseTypeLive    = 0, //大班模式                (非互动直播)
    TalkfunCourseTypeInteractiveLive = 1, // 大班互动模式    (互动直播)
    TalkfunCourseTypeStatusRTC = 2, //小班模式
    TalkfunCourseTypLowLatency = 3, //大班低延时模式
}TalkfunCourseType;
typedef NS_ENUM(NSUInteger, TalkfunMovieScalingMode) {
    TalkfunMovieScalingModeAspectFit = 0, ///< 图像适应屏幕，保持画面完整，但如果图像宽高比不同于屏幕宽高比，会有黑边的存在。
```


TalkfunMovieScalingModeFill = 1, // 图像铺满屏幕, 不留黑边, 如果图像宽高比不同于屏幕宽高比, 部分画面内容会被裁剪掉

```
};
@end
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
/**
 *      大班与小班的进入房间参数的模型
 */
@interface TalkfunSDKLiveInitParams : NSObject
/**【字段含义】访问令牌 [必填]
 */
@property (nonatomic, copy, nonnull) NSString* accessToken;
/** 课程模式 : (不设置的情况下默认为大班)
 */
@property (nonatomic, assign) TalkfunCourseType courseType;
@end

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
/**
 *      进入房间参数的模型
 */
@interface TalkfunSDKPlaybackInitParams : NSObject
/**【字段含义】访问令牌 [必填]
 */
@property (nonatomic, copy, nonnull) NSString* accessToken;
/**课程id
 */
@property (nonatomic, copy, nonnull) NSString* playbackID;
@end

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
/**
 *      生活直播的进入房间参数的模型
 */
@interface TalkfunSDKLifeLiveInitParams : NSObject
/**【字段含义】访问令牌 [必填],
 */
@property (nonatomic, copy, nonnull) NSString* accessToken;
@end

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
@interface TFLotteryStartUser : NSObject
/**发起抽奖人所在的组*/
@property (nonatomic, copy, nonnull) NSString* gid;
/**名字*/
```

```
@property (nonatomic, copy, nonnull)NSString* nickname;
/**角色
*/
@property(nonatomic,assign)TalkfunRtcRole role;
@end

/*抽奖开始的回调*/
@interface TFLotteryStart : NSObject
/**抽奖编号*/
@property (nonatomic, copy, nonnull)NSString *t;
/**发起抽奖人资料*/
@property (nonatomic, strong, nonnull)TFLotteryStartUser* user;
@end

@interface TFLotteryWinner : NSObject
/**中奖人的编号*/
@property (nonatomic, copy, nonnull)NSString* xid;
/**中奖人名字*/
@property (nonatomic, copy, nonnull)NSString* nickname;
/**发起抽奖人的名字*/
@property (nonatomic, copy, nonnull)NSString* launch_nickname;
/** 是不是自己*/
@property (nonatomic, assign)BOOL isMe;
@end

/*抽奖结束的回调*/
@interface TFLotteryStop : NSObject
@property (nonatomic, copy, nonnull)NSString* gid;
/**是否显示抽奖结果*/
@property (nonatomic, assign)BOOL pub;
@property (nonatomic, copy, nonnull)NSString* lotery_id;
//中奖人列表
@property (nonatomic, strong)NSMutableArray <TFLotteryWinner *>* _Nullable winnerList;
/**奖品名称*/
@property (nonatomic, copy, nonnull)NSString*prize_name;
@end

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

/**主播回复的内容*/
@interface TFQuestionAnswer : NSObject
@property (nonatomic, copy, nonnull) NSString* status;
@property (nonatomic, copy, nonnull) NSString* sn;
@property (nonatomic, copy, nonnull) NSString* qid;
/**回复的编号*/
@property (nonatomic, copy, nonnull) NSString* replyId;
@property (nonatomic, copy, nonnull) NSString* uid;
@property (nonatomic, copy, nonnull) NSString* liveid;
/**直播间是时间戳 例:1618036620
    点播间是秒数:60
*/
@property (nonatomic, copy, nonnull)NSString* time;
```

```

@property (nonatomic, copy, nonnull)NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull)NSString* avatar;
@property (nonatomic, copy, nonnull)NSString* replies;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* nickname;
@property (nonatomic, copy, nonnull) NSString* course_id;
@property (nonatomic, copy, nonnull) NSString* xid;
/**内容*/
@property (nonatomic, copy, nonnull) NSString* content;
@end

/*提问的回调*/
@interface TFQuestionData : NSObject
@property (nonatomic, copy, nonnull) NSString* course_id;
@property (nonatomic, copy, nonnull) NSString* xid;
/*提问的编号*/
@property (nonatomic, copy, nonnull) NSString* qid;
@property (nonatomic, copy, nonnull) NSString* uid;
/**回复的编号*/
@property (nonatomic, copy, nonnull) NSString* replyId;
@property (nonatomic, copy, nonnull) NSString* liveid;
/**主播回复的内容*/
@property (nonatomic, strong)NSMutableArray <TFQuestionAnswer *> _Nullable answer;
/**内容*/
@property (nonatomic, copy, nonnull) NSString* content;
/**直播间是时间戳 例:1618036620
    点播间是秒数:60
    */
@property (nonatomic, copy, nonnull)NSString* time;
@property (nonatomic, copy, nonnull)NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull)NSString* avatar;
@property (nonatomic, copy, nonnull)NSString* replies;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* nickname;
@property (nonatomic, copy, nonnull) NSString* sn;
@property (nonatomic, copy, nonnull) NSString* status;
/*chat:1 */
@property (nonatomic, copy, nonnull) NSString*chat;
/*删除提问*/
@property (nonatomic, copy, nonnull) NSString * deleteQuestionID;
@end
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//网络
@interface TFCurrentNetwork: NSObject
//14.145.136.104

```

```
@property(nonatomic, strong)NSString * _Nonnull ip;
//电信
@property(nonatomic, strong)NSString * _Nonnull isp;
//中国广东广州
@property(nonatomic, strong)NSString * _Nonnull location;
@end

//-----网络线路
@interface TFLiveLine: NSObject
//中国电信
@property(nonatomic, strong)NSString * _Nonnull name;
//telecom
@property(nonatomic, strong)NSString * _Nonnull key;
@end

/*网络线路组*/
@interface TFLiveLineData: NSObject
//线路列表
@property (nonatomic, strong)NSMutableArray <TFLiveLine *> _Nullable lineList;
@end

/*直播线路管理者*/
@interface TFLiveLineManager: NSObject
/*直播的线路组
*/
@property (nonatomic, strong)NSMutableArray <TFLiveLineData *> _Nullable data;
/*当前网络状态*/
@property(nonatomic, strong)TFCurrentNetwork * _Nullable network;
/*获取网络是否成功 code: 0 代表获取成功
*/
@property(nonatomic, assign)NSInteger code;
/*提示语*/
@property(nonatomic, strong)NSString* _Nonnull msg;
@end

@interface TFPlaybackLineData: NSObject
/*线路1 线路2*/
@property(nonatomic, strong)NSString* _Nonnull name;
/*索引*/
@property(nonatomic, assign)NSInteger index;
/*当前线路选中*/
@property(nonatomic, assign)BOOL current;
@end

/*回放或伪直播线路管理者*/
@interface TFPlaybackLineManager: NSObject
/*获取网络是否成功 code: 0 代表获取成功
*/
@property(nonatomic, assign)NSInteger code;
/*提示语*/
@property(nonatomic, strong)NSString* _Nonnull msg;
/*回放的线路组
```

```
*/
@property (nonatomic, strong)NSMutableArray <TFPlaybackLineData *>* _Nullable data;
@end

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/**
 * 公告
 */
@interface TFAnnouncement : NSObject
///【字段含义】内容
@property (nonatomic, copy, nonnull) NSString* content;
///【字段含义】公告发起人
@property (nonatomic, copy, nonnull) NSString* nickname;
///【字段含义】内容 "2021-03-15 19:27"
@property (nonatomic, copy, nonnull) NSString* time;
@end

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/**
 * 滚动通知
 */
@interface TFAnnounceRoll : NSObject
///【字段含义】时长
@property (nonatomic, assign) NSInteger duration;
///【字段含义】内容
@property (nonatomic, copy, nonnull) NSString* content;
///【字段含义】当前时间 (:2020-12-24 11:28)
@property (nonatomic, copy, nonnull) NSString* time;
///【字段含义】超链接
@property (nonatomic, copy, nonnull) NSString *link;
/**名字*/
@property (nonatomic, copy, nonnull) NSString* nickname;
@end

@interface TFChat : NSObject
@property (nonatomic, assign) NSInteger enable;
@property (nonatomic, assign) NSInteger grant;
@end

@interface TFChatReply : NSObject
/*被@的人的名字*/
@property (nonatomic, copy, nonnull) NSString*nickname;
/*被@的人的xid*/
@property (nonatomic, copy, nonnull) NSString*xid;
/** 是不是自己*/
@property (nonatomic, assign)BOOL isMe;
@end

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```



```
@property (nonatomic, copy, nonnull)NSString * time;
///【字段含义】评论
@property (nonatomic, copy, nonnull)NSString * remark;
///【字段含义】发起签到人的名字
@property (nonatomic, copy, nonnull)NSString * nickname;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
///【字段含义】签到的持续时长
@property (nonatomic, assign)NSInteger duration;
///【字段含义】发起签到人的头像
@property (nonatomic, copy, nonnull)NSString * avatar;
@end
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/**
 *      结束签到的事件回调
 */
@interface TFSignEnd : NSObject
///【字段含义】 ID
@property (nonatomic, copy, nonnull)NSString * signId;
///【字段含义】 total
@property (nonatomic, copy, nonnull)NSString * total;
///【字段含义】 signTotal
@property (nonatomic, copy, nonnull)NSString * signTotal;
@end
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/**
 *      直播时长
 */
@interface TFLiveDuration : NSObject
///   年
@property (nonatomic, assign)NSInteger year;
///   月
@property (nonatomic, assign)NSInteger month;
///   时
@property (nonatomic, assign)NSInteger day;
///   年
@property (nonatomic, assign)NSInteger hour;
///   分
@property (nonatomic, assign)NSInteger minute;
///   秒
@property (nonatomic, assign)NSInteger second;
@end

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//学员已选择的内容列表
```

```
@interface TFop : NSObject
///【字段含义】 选项的标题
@property (nonatomic, copy, nonnull)NSString * op;
///【字段含义】 选项的人数
@property (nonatomic, copy, nonnull)NSString * opNum;
///【字段含义】 选项的百分比
@property (nonatomic, copy, nonnull)NSString * percent;
///【字段含义】 选中
@property(nonatomic,assign)BOOL selected;
@end

//发起投票者的个人数据
@interface TFVoteInfo : NSObject
///【字段含义】 "正确答案 A,B , C"
@property (nonatomic, copy, nonnull)NSArray * answer;
///【字段含义】 名字
@property (nonatomic, copy, nonnull)NSString * nickname;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
///【字段含义】 标题
@property (nonatomic, copy, nonnull)NSString * title;
///【字段含义】 "14:10",
@property (nonatomic, copy, nonnull)NSString * noticeTime;
///【字段含义】 图片模式的投票
@property (nonatomic, copy, nonnull)NSString * imageUrl;
///【字段含义】 标签
@property (nonatomic, strong)NSString * _Nullable label;
///【字段含义】 "209890283",
@property (nonatomic, strong)NSString * _Nullable xid;
///【字段含义】 投票的发起时间 ""2021-03-24 16:13:09"
@property (nonatomic, strong)NSString * _Nullable startTime;
///【字段含义】 投票的结束 endTime == 1624591186 , 投票未结束 endTime = 0
@property(nonatomic,strong)NSString* _Nullable endTime ;
@end

//投票时选项的内容列表
@interface TFVoteNewUser : NSObject
///【字段含义】 ["0,1"], 选中的 索引
@property (nonatomic, copy, nonnull)NSMutableArray * option;
///【字段含义】 uid
@property (nonatomic, copy, nonnull)NSString * uid;
///【字段含义】 "19:52:46",
@property (nonatomic, copy, nonnull)NSString * time;
///【字段含义】 "7379725
@property (nonatomic, copy, nonnull)NSString * userId;
///【字段含义】 投票唯一编码
@property (nonatomic, copy, nonnull)NSString * vid;
///【字段含义】 投票人的名字
@property (nonatomic, copy, nonnull)NSString * nickname;
///【字段含义】 内容
@property (nonatomic, copy, nonnull)NSString * msg;
@end
```



```

/**通过xid获取当前投票详情*/
@interface TFVoteData : NSObject
///【字段含义】1 == 单选 其它==多选
@property (nonatomic, assign)NSInteger optional;
///【字段含义】学员选择的内容列表
@property (nonatomic, strong)NSMutableArray <TFOp *>* _Nullable opList;
///【字段含义】发起投票者的个人数据
@property (nonatomic, strong)TFVoteInfo *_Nullable info;
///【字段含义】ID
@property (nonatomic, copy, nonnull)NSString * gid;
///【字段含义】当前投票的ID
@property (nonatomic, copy, nonnull)NSString * vid;
///【字段含义】signTotal
@property (nonatomic, copy, nonnull)NSString * t;
///【字段含义】(0 未投票 1 已投票)
@property(nonatomic, assign)BOOL voted;
///【字段含义】(是否公布答案)
@property(nonatomic, assign)BOOL isShow;
///【字段含义】投票时选项的内容列表
@property (nonatomic, strong)NSMutableArray <TFVoteNewUser *>* _Nullable userList;
@end

```

```

/**提交投票答案*/
@interface TFSubmit: NSObject
///【字段含义】投票的ID
@property (nonatomic, strong)NSString * _Nullable vid;
///【字段含义】投票时用到的答案数组 ["1",2] 就是A, B
@property (nonatomic, strong)NSMutableArray * _Nullable option;
@end

```

```

@interface TFKeyNote : NSObject
@property(nonatomic, assign) NSInteger status; //0是主播是主讲人, 1是用户是主讲人
@property(nonatomic, strong) NSString * term;
@property(nonatomic, strong) NSString * nickname;
@property(nonatomic, strong) NSString * xid;
@end

```

```

/**广播*/
@interface TFBroadcast : NSObject
/**内容*/
@property(nonatomic, strong) NSString * msg;
@property(nonatomic, strong) NSString * uniqid;
@property(nonatomic, strong) NSString * __auto;
/**所在的组*/
@property(nonatomic, strong) NSString * gid;
/**时间戳 例:1618036620*/
@property(nonatomic, strong) NSString * time;
@end

```

```

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

/**网络速度*/
@interface TFNetworkSpeed : NSObject
@property(nonatomic,assign) TalkfunNetworkStatus  networkStatus;
//6.0KB/s
@property(nonatomic,strong) NSString * speed;

@end

@interface TFMemberParams : NSObject
@property(nonatomic,assign)NSInteger page;
@property(nonatomic,assign)NSInteger size;
@end

@interface TFMemberChat : NSObject
//开关
@property (nonatomic, assign)NSInteger enable;
@property (nonatomic, assign)NSInteger grant;
@end

@interface TFMemberVoice : NSObject
//开关
@property (nonatomic, assign)NSInteger enable;
@property (nonatomic, assign)NSInteger grant;
@end

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/**
 * 成员信息
 */
@interface TFMember : NSObject
///【字段含义】成员名字
@property (nonatomic, copy, nonnull) NSString* nickname;
///【字段含义】成员头像
@property (nonatomic, copy, nonnull) NSString* avatar;
@property (nonatomic, copy, nonnull) NSString* gender;
@property (nonatomic, copy, nonnull) NSString* uid;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**{
  聊天权限
"enable" : 1,
"grant" : 0
}
*/
@property (nonatomic,strong) TFMemberChat*chat;
/**{
  语音权限
"enable" : 1,
"grant" : 0
}
*/

```

```
@property (nonatomic, strong) TFMemberVoice*voice;
@property (nonatomic, copy, nonnull) NSString* a;
@property (nonatomic, copy, nonnull) NSString* xid;
/**成员所在分组*/
@property (nonatomic, copy, nonnull) NSString* gid;
@property (nonatomic, copy, nonnull) NSString* term;
/**成员是不是机器人 默认:NO*/
@property (nonatomic, assign)NSInteger robot;

//标记status;

@property (nonatomic, assign) TalkfunApplyStatus status;
//涂鸦
@property (nonatomic, assign) BOOL drawPower;

@end

/**聊天配置详情*/
@interface TFMod_text_liveConfigChat : NSObject
//聊天
@property(nonatomic,strong) NSString * name;
//开关
@property(nonatomic,assign) BOOL enable;
//聊天间隔
@property(nonatomic,strong) NSString * interval;

@end

/**问答配置详情*/
@interface TFMod_text_liveConfigQa : NSObject
//问答
@property(nonatomic,strong) NSString * name;
//开关
@property(nonatomic,assign) BOOL enable;
//问答限制 0不通过 1通过 默认不通过
@property(nonatomic,strong) NSString * logic;
//聊天间隔
@property(nonatomic,strong) NSString * interval;
@end

@interface TFMod_text_liveConfig : NSObject
//聊天配置
@property(nonatomic,strong) TFMod_text_liveConfigChat * chat;
//问答配置
@property(nonatomic,strong) TFMod_text_liveConfigQa * qa;
@end

@interface TFMod_text_live : NSObject
@property(nonatomic,strong) TFMod_text_liveConfig * config;
```

```
@end

@interface TFMod_vote_Config_visible : NSObject
@property(nonatomic,strong) NSString * name;
//开关
@property(nonatomic,assign) BOOL enable; //投票开关
@end

/**投票配置*/
@interface TFMod_vote_Config : NSObject

@property(nonatomic,strong) TFMod_vote_Config_visible * visible;
@end

/**投票模块*/
@interface TFMod_vote : NSObject

//开关
@property(nonatomic,assign) BOOL enable;

@property(nonatomic,strong) TFMod_vote_Config * config;
@end

@interface TFLiveScoreConfigValue : NSObject
//专业能力
@property(nonatomic,strong) NSString * name;
//种类
@property(nonatomic,assign) NSInteger sort;
//速度
@property(nonatomic,assign)NSInteger rate;
/* 选项 technologyScore effectScore*/
@property(nonatomic,strong) NSString *options;
@end

@interface TFLiveFlowerData : NSObject
@property(nonatomic,assign) NSString* time_interval;
@property(nonatomic,assign) NSString* uid;
@property(nonatomic,assign) NSString* bid;
@property(nonatomic,assign) NSString* roomid;
@property(nonatomic,assign) NSString* liveid;
//花朵总数
@property(nonatomic,assign) NSInteger amount;
@property(nonatomic,assign) NSString* sendtime;
@property(nonatomic,assign) NSString* left_amount;
@property(nonatomic,assign) NSString* gid;
/**头像*/
@property(nonatomic,assign) NSString* avatar;
@property(nonatomic,assign) NSString* pass_time;
@property(nonatomic,assign) NSString* left_time;
```

```
@property (nonatomic, assign)TalkfunRtcRole role;
@property(nonatomic,assign) NSString* pid;
@property(nonatomic,assign) NSString* nickname;
@property(nonatomic,assign) NSString* time;
/**用户唯一标识符*/
@property(nonatomic,assign) NSString* xid;
/**课程id*/
@property(nonatomic,assign) NSString* course_id;
@end

/**评分*/
@interface TFLiveScoreConfig : NSObject
///【字段含义】投票时选项的内容列表
@property (nonatomic, strong)NSMutableArray <TFLiveScoreConfigValue *>* _Nullable valueList;
@end

/**主讲人*/
@interface TFZhujiang : NSObject
//开关
@property(nonatomic,assign) BOOL enable;
//状态
@property(nonatomic,assign)NSInteger status;

@property (nonatomic, copy, nonnull)NSString* xid;
@end

/**用户摄像头*/
@interface TFUserCamera : NSObject
//邀请连麦 0无 1有
@property(nonatomic,assign) BOOL invite;
//申请连麦 0无 1有
@property(nonatomic,assign)BOOL apply;
/**记录有没有开启连麦权限*/
@property(nonatomic,assign)NSInteger lastDrawPower;
/**申请列表的数据*/
@property(nonatomic, strong)NSMutableArray<TFMember*>* applyList;
/**邀请列表的数据*/
@property(nonatomic, strong)NSMutableArray<TFMember*>* inviteList2;

/**讲台列表用户的数据*/

@property(nonatomic, strong)NSMutableArray<TalkfunRtcModel*>* upUsers;

@end
```

```
/**直播信息章节*/
@interface Mod_liveInfo_ConfigChapter : NSObject
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* name;
@end

/**直播信息的课程配置*/
@interface TMod_liveInfo_Config : NSObject

@property (nonatomic, strong)Mod_liveInfo_ConfigChapter* countdown;

//直播标题
@property (nonatomic, strong)Mod_liveInfo_ConfigChapter* title;

@property (nonatomic, strong)Mod_liveInfo_ConfigChapter* theme;

//直播时间

@property (nonatomic, strong)Mod_liveInfo_ConfigChapter* liveTime;

@end

/**直播信息的课程配置*/
@interface TMod_liveInfo_Live : NSObject

/* 开关 */
@property (nonatomic, assign) NSInteger enable;

/**直播信息的课程配置*/
@property(nonatomic,strong)TMod_liveInfo_Config *config;

@end

/**直播互动奖杯配置*/
@interface TMod_zhuboaction_Cup : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic,strong) NSString * name;

@end

/**直播互动投票配置*/
@interface TMod_zhuboaction_Vote : NSObject
```

```
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动通知配置*/
@interface TFMd_zhuboaction_Notice:NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动点名配置*/
@interface TFMd_zhuboaction_Rollcall:NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动计时器配置*/
@interface TFMd_zhuboaction_Timer:NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动公告配置*/
@interface TFMd_zhuboaction_Announcement:NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动广播配置*/
@interface TFMd_zhuboaction_Broadcast:NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动抢答器配置*/
```

```
@interface TFMod_zhuboaction_Responder: NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动转盘配置*/
@interface TFMod_zhuboaction_Turntable: NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动转盘配置*/
@interface TFMod_zhuboaction_Lottery: NSObject
/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**直播互动信息配置*/
@interface TFMod_zhuboaction_Config : NSObject
/**奖杯*/
@property (nonatomic, strong) TFMod_zhuboaction_Cup * cup; //
/**投票*/
@property (nonatomic, strong) TFMod_zhuboaction_Vote * vote; //
/**通知*/
@property (nonatomic, strong) TFMod_zhuboaction_Notice * notice; //
/**公告*/
@property (nonatomic, strong) TFMod_zhuboaction_Announcement * announcement; //
/**广播*/
@property (nonatomic, strong) TFMod_zhuboaction_Broadcast * broadcast;
/**点名*/
@property (nonatomic, strong) TFMod_zhuboaction_Rollcall * rollcall;
/**抢答器*/
@property (nonatomic, strong) TFMod_zhuboaction_Responder * responder;

/**定时器*/
@property (nonatomic, strong) TFMod_zhuboaction_Timer * timer;

/**转盘*/
@property (nonatomic, strong) TFMod_zhuboaction_Turntable * turntable;
/**抽奖*/
@property (nonatomic, strong) TFMod_zhuboaction_Lottery * lottery;

@end
```



```
/**直播互动信息配置*/
@interface TMod_zhuboaction_live : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

/**直播互动配置*/
@property(nonatomic, strong)TMod_zhuboaction_Config *config;

@end

/**课前聊天配置*/
@interface TMod_beforeclass_chat : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**课前信息配置*/
@interface TMod_beforeclass_Config : NSObject

@property(nonatomic, strong)TMod_beforeclass_chat * chat;
@end

/**课前互动信息配置*/
@interface TMod_beforeclass_live : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

/**课前互动配置*/
@property(nonatomic, strong)TMod_beforeclass_Config *config;

@end

/**课后聊天配置*/
@interface TMod_afterclass_chat : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

@property (nonatomic, strong) NSString * name;

@end

/**课后信息配置*/
@interface TMod_afterclass_Config : NSObject
```

```
@property(nonatomic, strong)TFMod_afterclass_chat * chat;
@end

/**课后互动信息配置*/
@interface TFMod_afterclass_live : NSObject

/* 开关 */
@property (nonatomic, assign) BOOL enable;

/**课后互动配置*/
@property(nonatomic, strong)TFMod_afterclass_Config *config;

@end

/**直播的课程配置*/
@interface TFLiveRoomInfo : NSObject

/*是否显示访客人数*/
@property (nonatomic, assign)BOOL visitors;
/*YES:单体禁言 NO:单体解除禁言 */
@property (nonatomic, assign)BOOL isChatDisable;
/*YES:全体禁言 NO:全体解除禁言 */
@property (nonatomic, assign)BOOL isDisableAll;
/*摄像头开关获取 */
@property (nonatomic, assign)BOOL isCamera;
/*同时在线人数*/
@property (nonatomic, assign)NSInteger peakConcurrentUsers;
/*在线的客服列表*/
@property (nonatomic, strong)NSMutableArray<TFMember *> _Nullable serviceList;

@property(nonatomic, strong) TFMod_text_live * mod_text_live;
/* 课后互动信息 */
@property(nonatomic, strong) TFMod_afterclass_live * mod_afterclass_live;
/* 课前互动信息 */
@property(nonatomic, strong) TFMod_beforeclass_live * mod_beforeclass_live;
/* 直播互动信息 */
@property(nonatomic, strong) TFMod_zhuboaction_live * mod_zhuboaction_live;
/* 用户摄像头信息 */
@property(nonatomic, strong) TFUserCamera *usercamera;
/* 直播信息 */
@property(nonatomic, strong) TFMod_liveInfo_Live * mod_liveinfo_live;

/* 获取当前用户个人数据 */
@property (nonatomic, strong)TFMember * myInfo;
/*评分系统是否开启*/
//@property(nonatomic, assign)BOOL evaluate;
/* 评分的数据 */
@property(nonatomic, strong)TFLiveScoreConfig *scoreConfig;
/**获取当前花朵总数*/
@property(nonatomic, assign) NSInteger flowerTotal;
```

```
/**课程是伪直播*/
@property(nonatomic,assign) BOOL isVodLive;
/* 获取主播个人数据 */
@property (nonatomic,strong)TFMember * zhuboInfo;
/* 获取课程标题 */
@property (nonatomic,strong)NSString * title;

/* 获取主讲人数据 */
@property (nonatomic,strong)TFZhujiang* zhujiang;
/*课程开始的时间戳 */
@property (nonatomic,assign)NSInteger startTime;
/*课程结束的时间戳 */
@property (nonatomic,assign)NSInteger end_time;
///【字段含义】暖场视频
@property (nonatomic, strong)NSString* _Nullable videoUrl;
/*在线的机器人列表*/
@property (nonatomic, strong)NSMutableArray<TFMember *> _Nullable robotList;
/**投票模块*/
@property(nonatomic,strong) TFMod_vote * mod_vote;

@end

/**课程配置*/
@interface TFLiveModeChange : NSObject
/* 当前模式 */
@property(nonatomic,assign) TalkfunLiveMode currentMode;
/* 上一次的模式 */
@property (nonatomic,assign)TalkfunLiveMode beforeMode;
@end

/**点播水印*/
@interface TFMod_theftproof : NSObject
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**xid*/
@property (nonatomic, copy, nonnull)NSString* xid;
@end

/**点播聊天*/
@interface TFMod_playbackinfo_playbackConfigChat : NSObject
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* name;
@end

/**点播章节*/
@interface TFMod_playbackinfo_playbackConfigChapter : NSObject
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* name;
@end
```

```
/**点播提问*/
@interface TFMod_playbackinfo_playbackConfigQa : NSObject
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* name;
@property (nonatomic, copy, nonnull)NSString* logic;
@property (nonatomic, copy, nonnull)NSString* interval;
@end

/**点播的 聊天---章节---提问 */
@interface TFMod_playbackinfo_playbackConfig : NSObject
/* 聊天 */
@property (nonatomic, strong)TFMod_playbackinfo_playbackConfigChat *chat;
/* 章节 */
@property (nonatomic, strong)TFMod_playbackinfo_playbackConfigChapter *chapter;
/* 提问 */
@property (nonatomic, strong)TFMod_playbackinfo_playbackConfigQa *qa;
@end

@interface TFMod_playbackinfo_playback : NSObject
/**回放信息*/
@property (nonatomic, copy, nonnull)NSString* name;
/* 开关 */
@property (nonatomic, assign) NSInteger enable;
/**点播的聊天与提问的配置开关*/
@property (nonatomic, strong)TFMod_playbackinfo_playbackConfig *config;
@end

/**点播的课程配置*/
@interface TFPlaybackRoomInfo : NSObject
/* 水印 */
@property (nonatomic, strong)TFMod_theftproof *mod_theftproof;
/* 回放信息*/
@property (nonatomic, strong)TFMod_playbackinfo_playback *mod_playbackinfo_playback;

//课程总时长
@property (nonatomic, assign) CGFloat totalTime;
@end

NS_ASSUME_NONNULL_END

@interface TFPlaybackAlbum : NSObject
//1768563 id = playbackID
@property (nonatomic, copy, nonnull)NSString *playbackID;
//计算机一级-精讲6
@property (nonatomic, copy, nonnull)NSString *title;
//https://s2.talk-fun.com/4/live/563/59177/mtc7odu7mw/thumb-80x60.jpg
```

```
@property (nonatomic, copy, nonnull)NSString *img_small;
//gYyYjM5IjMxQDOE1T0xQERBJTRzIkI6ICZpVnIsIiclNXdiojIlx2byJye
@property (nonatomic, copy, nonnull)NSString *access_token;
// 课程时长:秒数 161
@property (nonatomic, assign)NSInteger duration;
/**当前选中的专辑*/
@property(nonatomic,assign)BOOL select;
@end

//章节page
@interface TFPlaybackChapterPage : NSObject
/**时间 例:55 */
@property (nonatomic, assign)CGFloat starttime ;
/**图像*/
@property (nonatomic, copy, nonnull)NSString* thumbnail;
/**页面名字*/
@property (nonatomic, copy, nonnull)NSString* course;
/**当前页码*/
@property (nonatomic, assign)NSInteger page;
@end

//章节page
@interface TFPlaybackChatData : NSObject
@property (nonatomic, copy, nonnull) NSString* uid;
@property (nonatomic, copy, nonnull) NSString* a;
//1622119692
@property (nonatomic, copy, nonnull) NSString* timestamp;
//7276
@property (nonatomic, copy, nonnull) NSString* starttime;
/**时间 例:55 */
@property (nonatomic, copy, nonnull) NSString* time;
@property (nonatomic, copy, nonnull) NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull) NSString* avatar;
@property (nonatomic, copy, nonnull) NSString* cmd;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**名字*/
@property (nonatomic, copy, nonnull) NSString* nickname;
/**内容*/
@property (nonatomic, copy, nonnull) NSString * msg;
@property (nonatomic, copy, nonnull) NSString* status;
@property (nonatomic, copy, nonnull) NSString* xid;
@end

/**主播回复的内容*/
@interface TFPlaybackQuestionAnswer : NSObject
//@property (nonatomic, copy, nonnull) NSString* status;
@property (nonatomic, copy, nonnull) NSString* sn;
@property (nonatomic, copy, nonnull) NSString* qid;
```

```

/**回复的编号*/
//@property (nonatomic, copy, nonnull) NSString* replyId;
@property (nonatomic, copy, nonnull) NSString* uid;
/*点播间是秒数:60*/
@property (nonatomic, copy, nonnull) NSString* startTime;
//@property (nonatomic, copy, nonnull) NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull) NSString* avatar;
/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**名字*/
@property (nonatomic, copy, nonnull) NSString* nickname;
@property (nonatomic, copy, nonnull) NSString* xid;
/**内容*/
@property (nonatomic, copy, nonnull) NSString* content;
@end

/////点播的提问
@interface TFPlaybackQuestionData : NSObject

@property (nonatomic, copy, nonnull) NSString* xid;
/*提问的编号*/
@property (nonatomic, copy, nonnull) NSString* qid;
@property (nonatomic, copy, nonnull) NSString* uid;

/**主播回复的内容*/
@property (nonatomic, strong)NSMutableArray <TFPlaybackQuestionAnswer *> _Nullable answer;
/**内容*/
@property (nonatomic, copy, nonnull) NSString* content;
/**
    点播间是秒数:60
*/
@property (nonatomic, copy, nonnull) NSString* startTime;
//@property (nonatomic, copy, nonnull) NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull) NSString* avatar;

/**角色*/
@property (nonatomic, assign)TalkfunRtcRole role;
/**名字*/
@property (nonatomic, copy, nonnull) NSString* nickname;
@property (nonatomic, copy, nonnull) NSString* sn;
//@property (nonatomic, copy, nonnull) NSString* status;

@end

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
/**
*   主播的资料

```

```
*/
@interface TFZhubo : NSObject
///【字段含义】名字
@property (nonatomic, copy, nonnull) NSString* nickname;
///【字段含义】头像
@property (nonatomic, copy, nonnull) NSString* avatar;
@end
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-08-26 17:26:23

直播接口说明

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

进入房间

```
//配置房间参数
TalkfunSDKLiveInitParams *params = [[TalkfunSDKLiveInitParams alloc]init];
params.accessToken = access_token;
params.courseType = TalkfunCourseTypeInteractiveLive;

//TODO:初始化SDK-----
self.talkfunSDK = [TalkfunSDKLive sharedInstance];
//进入房间
[self.talkfunSDK enterRoom:params];
```

进入房间的事件回调

- TalkfunSDKLiveDelegate

```
/**@name 进入房间的事件回调
 * - 如果加入成功,code 会是一个正数 (code > 0),代表加入房间成功)。 error 为nil
 * - 如果加入失败,code 会是一个负数 (code < 0),代表进房失败的错误码。 error 有值
 * 调用刷新refreshData: 也会重新回调
 * 进房失败的错误码含义请参见[错误码 TalkfunSDK 文件的TalkfunError 枚举
 */
- (void)onEnterRoom:(NSInteger)code result:(TFError*_Nonnull)error
```

-----下面的代码为过期的方法,不建议使用-----

房间

房间初始化: TALKFUN_EVENT_ROOM_INIT

@参数说明:

参数	类型	描述
action	NSString	直播状态
roomInfo	NSDictionary	房间数据

roomInfo说明:

me: 当前用户的信息

参数	类型	描述
roomid	NSNumber	房间ID
uid	NSString	合作方用户ID
xid	NSNumber	欢拓用户ID
avatar	NSString	用户头像地址
nickname	NSString	用户昵称

Zhubo : 老师的信息

参数	类型	描述
nickname	NSString	老师昵称
intro	NSString	老师简介
flower	NSString	鲜花数量
p_40	NSString	40x40尺寸头像
p_150	NSString	150x150尺寸头像

Announce : 公告内容 ##### 公告:

参数	类型	描述
content	NSString	公告内容
time	NSString	公告发布时间

滚动通知:

参数	类型	描述
content	NSString	通知内容
link	NSString	链接
duration	NSString	设置的滚动时长
time	NSString	通知发布时间

广播:

参数	类型	描述
message	NSString	广播信息
__auto	NSNumber	是否自动发给后来者
uniqid	NSString	广播唯一标识
t	NSNumber	时间戳

```
实例: [self.talkfunSDK on:TALKFUN_EVENT_ROOM_INIT callback:^(id obj) { //返回数据 { action = wait; mode = 2; roomInfo = { announce = { notice = { content = "5120\n"; id = 97; "partner_id" = 20; roomid = 550481; time = "2016-03-03 20:05"; }; roll = ""; }; initEvent = ( { args = { "__auto" = 1; message = "854321.0"; t = 1457086318; uniqid = 56d95f6e08367; }; cmd
```

```
= broadcast; } ); me = { avatar = ""; nickname = rrrr; pid = 20; role = user; roomid =  
550481; uid = 1185252; xid = 337861; }; zhubo = { flower = 451; intro = ""; liveid = 1269050;  
nickname = "\U5146\U80fd"; "p_150" = "http://static-1.talk-  
fun.com/open/maituo/static/css/img/_150.png"; "p_40" = "http://static-1.talk-  
fun.com/open/maituo/static/css/img/_40.png"; rid = 550481; }; }; } }];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:45:00

视频容器

- 只需要在talkfunSDK初始化后传入一个视频容器即可
- 播放时sdk会生成一个视频图添加到该容器
- 也可以在talkfunSDK初始化后,通过- (void)exchangePPTAndCameraContainer 交换ppt容器和摄像头容器

```
- (void)configureCameraContainerView:(UIView *)cameraContainerView;
```

调用示例：

把视频容器给SDK（要显示摄像头区域的必须部分）

```
[self.talkfunSDK configureCameraContainerView:self.cameraView];
```

摄像头显示： TALKFUN_EVENT_CAMERA_SHOW

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CAMERA_SHOW callback:^(id obj) {  
    //可以做显示摄像头的操作  
}]
```

摄像头隐藏： TALKFUN_EVENT_CAMERA_HIDE

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CAMERA_HIDE callback:^(id obj) {  
    //可以做隐藏摄像头的操作  
}]
```

桌面分享开始： TALKFUN_EVENT_DESKTOP_START

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_DESKTOP_START callback:^(id obj) {  
    //没有返回数据  
}];
```

桌面分享结束： TALKFUN_EVENT_DESKTOP_STOP

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_DESKTOP_STOP callback:^(id obj) {
//没有返回数据
}];
```

桌面分享暂停： TALKFUN_EVENT_DESKTOP_PAUSE

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_DESKTOP_PAUSE callback:^(id obj) {
//没有返回数据
}];
```

切换视频模式： TALKFUN_EVENT_MODE_CHANGE

当由课件模式或桌面分享/视频插播模式切换时，会调用

@参数说明：字典，字典里面数据如下：

```
、
    视频模式的枚举

typedef enum {
    TalkfunLiveModeVideo    = 0,
    TalkfunLiveModeDesktop = 2,
    TalkfunLiveModeRTC= 3
}TalkfunLiveMode;

...

```

参数	类型	描述
beforeMode	NSString	当前状态 (TalkfunLiveMode 枚举值)
beforeMode	NSString	本来的状态 (TalkfunLiveMode 枚举值)

调用示例： [self.talkfunSDK on:TALKFUN_EVENT_MODE_CHANGE callback:^(id obj) {

```
TalkfunLiveMode mode = [obj[@"currentMode"] intValue];
TalkfunLiveMode mode2 = [obj[@"beforeMode"] intValue];

if (mode != mode2) {
    weakSelf.tipsLabel.text = @"切换中...";
    [weakSelf.pptView addSubview:weakSelf.tipsLabel];
}

```

});

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

配置画板

- - (void)configurePPTContainerView:(UIView *)pptContainerView

调用示例：

```
[self.talkfunSDK configurePPTContainerView:self.pptContainerView];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:05

直播间状态

- TalkfunSDKLiveDelegate

```
- (void)onLiveStatus:(TalkfunLiveStatus)status;
```

*TalkfunLiveStatus 参数说明：

```
typedef enum {  
    TalkfunLiveStatusWait = 0, //直播未开始  
    TalkfunLiveStatusStart = 1, //直播开始  
    TalkfunLiveStatusStop = 2, //直播结束  
}TalkfunLiveStatus;
```

-----下面的代码为过期的方法,不建议使用-----

直播状态

直播开始: TALKFUN_EVENT_LIVE_START

@参数说明：

参数	类型	描述
title	NSString	直播主题

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_LIVE_START callback:^(id obj) {  
    //返回数据  
    {"title":"新课程"}  
}];
```

直播结束: TALKFUN_EVENT_LIVE_STOP

@参数说明：//没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_LIVE_STOP callback:^(id obj) {  
  
}];
```


当前用户进入房间触发

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onMemberJoinMe:(TFMember*_Nullable)result
```

其它人进入房间触发

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onMemberJoinOther:(TFMember*_Nullable)result
```

有人退出房间

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onMemberLeave:(TFMember*_Nullable)result
```

同时在线人数 (真实人+虚拟人员 的总数)

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onPeakConcurrentUsers:(NSInteger)total
```

相互挤下线

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onMemberForceout
```

当前用户被踢出房间的事件回调

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onMemberKick
```

- TFMember

```

//成员信息
@interface TFMember : NSObject
///【字段含义】成员名字
@property (nonatomic, copy, nonnull) NSString* nickname;
///【字段含义】成员头像
@property (nonatomic, copy, nonnull) NSString* avatar;
@property (nonatomic, copy, nonnull) NSString* gender;
@property (nonatomic, copy, nonnull) NSString* uid;
/**角色
extern NSString * const TalkfunMemberRoleSpadmin; //超级管理员(老师)
extern NSString * const TalkfunMemberRoleAdmin; //管理员(助教)
extern NSString * const TalkfunMemberRoleUser; //普通用户(学生)
extern NSString * const TalkfunMemberRoleGuest; //游客
*/
@property (nonatomic, copy, nonnull) NSString* role;
/**{
"enable" : 1,
"grant" : 0
}
*/
@property (nonatomic, copy, nonnull) NSDictionary*chat;
/**{
"enable" : 1,
"grant" : 0
}
*/
@property (nonatomic, copy, nonnull) NSDictionary*voice;
@property (nonatomic, copy, nonnull) NSString* a;
@property (nonatomic, copy, nonnull) NSString* xid;
/**成员所在分组*/
@property (nonatomic, copy, nonnull) NSString* gid;
@property (nonatomic, copy, nonnull) NSString* term;

@end

```

-----下面的代码为过期的方法,不建议使用-----

新用户进入直播间: **TALKFUN_EVENT_MEMBER_JOIN_OTHER**

@参数说明:

member

参数	类型	描述
xid	NSNumber	欢拓用户ID
uid	NSString	合作方用户ID
nickname	NSString	用户昵称
role	NSString	用户角色
gender	NSNumber	用户性别

Total:用户总数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_MEMBER_JOIN_OTHER callback:^(id res) {
//返回数据
{
    member = {
        avatar = "";
        chat = {
            enable = 1;
            grant = 0;
        };
        gender = 0;
        nickname = rrr;
        role = user;
        uid = 2277843;
        voice = {
            enable = 1;
            grant = 0;
        };
        xid = 346440;
    };
    total = 3;
}
}];
```

在线用户总数： TALKFUN_EVENT_MEMBER_TOTAL

@参数说明：

参数	类型	描述
total	NSNumber	在线人数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_MEMBER_TOTAL callback:^(id obj) {
//{"total":2}
}];
```

用户离开房间: TALKFUN_EVENT_MEMBER_LEAVE

@参数说明:

参数	类型	描述
member	NSDictionary	走掉的人的信息
total	NSNumber	剩余总人数

调用示例:

```
[self.talkfunSDK on:TALKFUN_EVENT_MEMBER_LEAVE callback:^(id obj) {
//返回数据
{
    nickname = rrr;
    total = 2;
    uid = 2277843;
    xid = 346440;
}
}];
```

member说明:

参数	类型	描述
xid	NSNumber	欢拓用户ID
uid	NSString	合作方用户ID
nickname	NSString	用户昵称
role	NSString	用户角色
gender	NSNumber	用户性别
voice	NSDictionary	是否可以说话(1是可以)
chat	NSDictionary	是否可以聊天(1是可以)

踢人: TALKFUN_EVENT_MEMBER_KICK

@参数说明:

参数	类型	描述
xid	NSNumber	欢拓用户ID
nickname	NSString	欢拓用户名
role	NSString	用户角色
voice	NSDictionary	是否可以说话(1是可以)
chat	NSDictionary	是否可以聊天(1是可以)
video	NSDictionary	是否可以看直播(1是可以)

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_MEMBER_KICK callback:^(id obj) {
//返回数据
{
    chat = {
        enable = 0;
        grant = 1;
    };
    nickname = rrr;
    role = user;
    video = {
        enable = 1;
        grant = 0;
    };
    voice = {
        enable = 1;
        grant = 0;
    };
    xid = 346440;
}
}];
```

被人强迫下线： TALKFUN_EVENT_MEMBER_FORCEOUT

@参数说明：

参数	类型	描述
uid	NSString	合作方用户ID
xid	NSNumber	欢拓用户ID
role	NSString	用户角色
state	NSNumber	状态
nickname	NSString	欢拓用户名
gender	NSNumber	用户性别
sessionid	NSString	服务器返回的标志
voice	NSDictionary	是否可以说话(1是可以)
video	NSDictionary	是否可以看直播(1是可以)
chat	NSDictionary	是否可以聊天(1是可以)
heartbeatTime	NSNumber	被强迫下线时间
roomid	NSNumber	房间ID
lastTime	NSNumber	进入房间时间

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_MEMBER_FORCEOUT callback:^(id obj) {  
//返回数据  
{  
    attr =    (  
    );  
    avatar = "";  
    chat =    {  
        enable = 1;  
        grant = 0;  
    };  
    gender = 0;  
    heartbeatTime = 1457088463;  
    nickname = rrrr;  
    regtime = 1457087820;  
    role = user;  
    roomid = 550481;  
    sessionId = "2697a65d-1c8b-4095-9925-9316eb825aba";  
    sid = "a5t5o14-rlvRsvLeAAN-";  
    state = 1;  
    uid = 1185252;  
    video =    {  
        enable = 1;  
        grant = 0;  
    };  
    voice =    {  
        enable = 1;  
        grant = 0;  
    };  
    xid = 337861;  
    }  
}}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:45:00

发送聊天

调用示例：

```
[self.talkfunSDK sendChat:self.chatBarView.chatBar.text callback:^(NSInteger code, TFChatData * _Nullable model, TFError * _Nullable error) {

    if (code == 0 ) {
        }else{

    }

}];
```

接收别人的聊天事件回调

- TalkfunSDKLifeLiveDelegate 或 TalkfunSDKLiveDelegate

调用示例：

```
- (void)onDidReceiveText:(TFChatData*_Nullable)result
{

}

}
```

-----下面的代码为过期的方法,不建议使用-----

发送聊天： TALKFUN_EVENT_CHAT_SEND

调用示例：

```
[talkfunSDK emit:TALKFUN_EVENT_CHAT_SEND parameter:@{@"msg":@"hello world!" } callback:^(id obj) {
{
//返回数据
    avatar = "";
    chat = {
        enable = 1;
        grant = 0;
    };
    gender = 0;
    msg = "\U65a4\U65a4\U8ba1\U8f83\U4e86";
    nickname = rrr;
    role = user;
    time = 1457087833;
    uid = 2277843;
    xid = 346440;
    }

}];
```

参数说明：

参数	类型	描述
msg	NSString	发送的消息内容

私聊： TALKFUN_EVENT_CHAT_PRIVATE

调用示例：

```
[talkfunSDK emit:TALKFUN_EVENT_CHAT_PRIVATE parameter:@{@"xid":1234,@"msg":@"hello world!"} callback:^(id obj) {
{
//返回数据
avatar = "";
chat = {
enable = 1;
grant = 0;
};
gender = 0;
msg = "\U65a4\U65a4\U8ba1\U8f83\U4e86";
nickname = rrr;
role = user;
time = 1457087833;
uid = 2277843;
xid = 346440;
}
}];
```

参数说明：

参数	类型	描述
xid	int	指定发送者的欢拓用户ID
msg	NSString	发送的消息内容

接收聊天： TALKFUN_EVENT_CHAT_SEND

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CHAT_SEND callback:^(id res) {
//返回数据
{
"xid":190045,
"uid":"2912763",
"nickname":"Superman",
"role":"user",
"gender":0,
"avatar":"",
"msg":"你好",
"time":1450334815
}
}
}];
```

参数说明：

参数	类型	描述
xid	NSNumber	用户唯一ID
uid	NSString	合作方用户ID
nickname	NSString	用户昵称
role	NSString	用户角色 (user/admin/guest/watch , 分别对应普通用户/管理员(助教)/游客/直播监考)
gender	NSNumber	用户性别
avatar	NSString	用户头像地址
msg	NSString	消息内容
time	NSNumber	时间戳(从1970-01-01到现在的秒数)

私聊: TALKFUN_EVENT_CHAT_PRIVATE

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CHAT_PRIVATE callback:^(id res) {
//返回数据
{
"xid":190045,
"uid":"2912763",
"nickname":"Superman",
"role":"user",
"gender":0,
"avatar":"",
"msg":"你好",
"time":1450334815
}
}
}];
```

参数说明：

参数	类型	描述
xid	NSNumber	用户唯一ID
uid	NSString	合作方用户ID
nickname	NSString	用户昵称
role	NSString	用户角色
gender	NSNumber	用户性别
avatar	NSString	用户头像地址
msg	NSString	消息内容
time	NSNumber	时间戳(从1970-01-01到现在的秒数)

禁言: TALKFUN_EVENT_CHAT_DISABLE

调用示例:

```
[self.talkfunSDK on:TALKFUN_EVENT_CHAT_DISABLE callback:^(id obj) {
//返回数据
{
"cmd":"chat:disable",
"args":
{
"xid":337861,
"nickname":"rrrr"
}
}
}];
```

参数说明:

参数	类型	描述
xid	NSNumber	用户ID
nickname	NSString	用户名字

Copyright Talkfun all right reserved , powered by Gitbook修订时间: 2021-05-25 16:45:00

发起提问

- - (void)enterAskQuestions:(NSString _Nullable)msg callback:(void (^_nullable) (NSInteger code, TFQuestionData __Nullable model ,TFError * _Nullable error))callback

```
调用示例 :
[self.talkfunSDK enterAskQuestions:self.setCenter.ask.textView.text callback:^(NSInteger code, TFQuestionData * _Nullable model, TFError * _Nullable error) {
    if (code ==0 ) {
        }
    }
};
```

接收别人的提问

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onQuestionAsk:(TFQuestionData*_Nonnull)result
```

主播回答了问题

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onQuestionReply:(TFQuestionData*_Nonnull)result
```

提问删除

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onQuestionDelete:(TFQuestionData*_Nonnull)result
```

提问审核通过

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onQuestionAudit:(TFQuestionData*_Nonnull)result
```

- 提问的sag

```

@interface TFQuestionData : NSObject
@property (nonatomic, copy, nonnull) NSString* course_id;
@property (nonatomic, copy, nonnull) NSString* xid;
/**提问的编号*/
@property (nonatomic, copy, nonnull) NSString* qid;
@property (nonatomic, copy, nonnull) NSString* uid;
/**回复的编号*/
@property (nonatomic, copy, nonnull) NSString* replyId;
@property (nonatomic, copy, nonnull) NSString* liveid;
/**主播回复的内容*/
@property (nonatomic, strong)NSMutableArray <TFQuestionAnswer *>* _Nullable answer;
/**内容*/
@property (nonatomic, copy, nonnull) NSString* content;
/**时间戳 例:1618036620*/
@property (nonatomic, copy, nonnull)NSString* time;
@property (nonatomic, copy, nonnull)NSString* gid;
/**头像链接*/
@property (nonatomic, copy, nonnull)NSString* avatar;
@property (nonatomic, copy, nonnull)NSString* replies;
/**角色
extern NSString * const TalkfunMemberRoleSpadmin; //超级管理员(老师)
extern NSString * const TalkfunMemberRoleAdmin; //管理员(助教)
extern NSString * const TalkfunMemberRoleUser; //普通用户(学生)
extern NSString * const TalkfunMemberRoleGuest; //游客
*/
@property (nonatomic, copy, nonnull)NSString* role;
/**名字*/
@property (nonatomic, copy, nonnull)NSString* nickname;
@property (nonatomic, copy, nonnull) NSString* sn;
@property (nonatomic, copy, nonnull) NSString* status;
/*chat:1 */
@property (nonatomic, copy, nonnull) NSString*chat;
/*删除提问*/
@property (nonatomic, copy, nonnull) NSString * deleteQuestionID;
@end
    
```

-----下面的代码为过期的方法,不建议使用-----

提问: TALKFUN_EVENT_QUESTION_ASK

@参数格式:

参数	类型	描述
msg	NSString	提问的内容

调用示例：

```
[talkfunSDK emit:TALKFUN_EVENT_QUESTION_ASK parameter:@{@"content":@"老师我有个问题!"} callback:^(id obj) {
{
content = 123;
nickname = "\U5146\U80fd";
qid = 40437;
replies = "<null>";
replyId = 40436;
role = spadmin;
time = 1457087721;
uid = "zb_12526";
xid = 300218;
}
}];
```

回复提问： TALKFUN_EVENT_QUESTION_REPLY

@参数格式：

参数	类型	描述
replyId	NSString	回复的提问ID
msg	NSString	提问的内容

调用示例：

```
[talkfunSDK emit:TALKFUN_EVENT_QUESTION_REPLY parameter:@{@"replyId":@"40437",@"msg":@"老师我有个问题!"} callback:^(id obj) {
{
content = 123;
nickname = "\U5146\U80fd";
qid = 40437;
replies = "<null>";
replyId = 40436;
role = spadmin;
time = 1457087721;
uid = "zb_12526";
xid = 300218;
}
}];
```

广播通知

接收提问： TALKFUN_EVENT_QUESTION_ASK

@参数说明：

参数	类型	描述
content	NSString	提问内容
xid	NSNumber	用户唯一ID
uid	NSString	合作方用户ID
qid	NSString	提问ID
nickname	NSString	用户昵称
role	NSString	角色
sn	NSNumber	提问顺序
time	NSNumber	时间戳(从1970-01-01到当前的秒数)

调用示例：

```
[talkfunSDK on: TALKFUN_EVENT_QUESTION_ASK callback:^(id res) {
//返回数据
{
content = "\U5065\U5065\U5eb7\U5eb7";
nickname = rrrr;
qid = 40436;
role = user;
sn = 3;
time = 1457087675;
uid = 1185252;
xid = 337861;
}
}];
```

接收提问的回复： TALKFUN_EVENT_QUESTION_REPLY

@参数说明：

参数	类型	描述
content	NSString	回复的内容
replyId	NSString	被回复的提问的ID
qid	NSString	回复的ID
nickname	NSString	回复者昵称
xid	NSNumber	欢拓用户ID
uid	NSString	合作方用户ID
role	NSString	用户角色
time	NSNumber	时间戳

调用示例：

```
[talkfunSDK on:TALKFUN_EVENT_QUESTION_REPLY callback:^(id res) {
//返回数据
{
    content = 123;
    nickname = "\U5146\U80fd";
    qid = 40437;
    replies = "<null>";
    replyId = 40436;
    role = spadmin;
    time = 1457087721;
    uid = "zb_12526";
    xid = 300218;
}
}];
```

提问列表： TALKFUN_EVENT_QUESTION_LIST

@参数说明：

参数	类型	描述
qid	NSString	提问ID
liveid	NSString	直播ID
uid	NSString	合作方用户ID
xid	NSNumber	欢拓用户ID
role	NSString	用户角色
nickname	NSString	用户昵称
content	NSString	提问内容
time	NSNumber	提问时间戳
replies	NSNumber	回复数
answer	NSArray	回复

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_QUESTION_LIST callback:^(id obj) {
{
    code = 0;
    data = {
        40434 = {
            content = 453453;
            liveid = 1269050;
            nickname = "\U5146\U80fd";
            qid = 40434;
            replies = 0;
            replyId = 0;
            role = spadmin;
            sn = 1;
            startTime = 3559;
            status = 0;
            time = 1457087092;
            uid = "zb_12526";
            xid = 300218;
        };
        40435 = {
            content = 123123123;
            liveid = 1269050;
            nickname = "\U5146\U80fd";
            qid = 40435;
            replies = 0;
            replyId = 0;
            role = spadmin;
            sn = 2;
            startTime = 3564;
            status = 0;
            time = 1457087097;
            uid = "zb_12526";
            xid = 300218;
        };
    };
};
}
}];
```

老师删除问题信息： TALKFUN_EVENT_QUESTION_DELETE

@参数说明：

参数	类型	描述
xid	NSNumber	用户ID
nickname	NSString	用户名字

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_QUESTION_DELETE callback:^(id obj) {  
    //返回数据  
    {"cmd":"chat:disable","args":{"xid":337861,"nickname":"rrrr"}}  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:45:00

投票开始的事件回调

- TalkfunSDKLiveDelegate

```
- (void)onVoteNew:(TFVoteData*_Nullable)result
```

投票结束的事件回调

- TalkfunSDKLiveDelegate

```
- (void)onVotePub:(TFVoteData*_Nullable)result
```

获取投票列表

- - (void)getVotingList:(void (^__nullable)(NSInteger code ,NSMutableArray __Nullable list ,TFError __Nullable error))callback

调用示例：

```
[self.talkfunSDK getVotingList:^(NSInteger code,NSMutableArray<TFVoteData *> *list, TFError *error) {  
  
    if (code == 0 ) {  
  
    }else{  
  
    }  
}];
```

获取投票ID详情

- - (void)getVotingDetails:(NSString__Nullable)vid callback:(void (^__nullable)(NSInteger code , TFVoteData __Nullable model,TFError *__Nullable error))callback

调用示例：

```
[self.talkfunSDK getVotingDetails:model.vid callback:^(NSInteger code, TFVoteData *model , TFError * __Nullable error) {  
    if (code == 0 ) {  
  
    }else{  
  
    }  
}];
```

确认投票

- (void)voteSubmit:(TFSubmit_Nullable)model callback:(void (^_nullable)(NSInteger code , NSError * _Nullable error))callback

调用示例：

```
TFSubmit *model = [[TFSubmit alloc]init];
model.vid = self.voteView.scrollView.result.vid;

[self.talkfunSDK voteSubmit:model callback:^(NSInteger code, NSError * _Nullable error)
] {
    if (code == 0) {
        }else{

    }
}];
```

-----下面的代码为过期的方法,不建议使用-----

投票开始: TALKFUN_EVENT_VOTE_NEW

@参数说明：

参数	类型	描述
vid	NSString	投票ID
optional	NSNumber	可选项数
opList	NSArray	选项列表
info	NSDictionary	投票发起信息

info 的主要参数说明

参数	类型	描述
title	NSString	不是图片投票时候的标题
label	NSString	图片投票时候的标题
imageUrl	NSString	图片地址（区分是否是图片投票、有值则为图片投票）
startTime	NSString	投票发起时间

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOTE_NEW callback:^(id obj) {
//返回数据
{
    opList = (          "发",          "个",          "好",          );
    info = {
        answer = "";
        nickname = "江如彬";
        role = "spadmin";
        title = "是的是的是的";
        noticeTime = "16:08";
        imageUrl = "";
        label = "是的 ";
        xid = 1185152;
        startTime = "2017-03-31 16:08:13";
        gid = 0;
    };
    vid = "18910";
    optional = 1;
}
}];
```

投票结束： TALKFUN_EVENT_VOTE_PUB

@参数说明：

参数	类型	描述
info	NSDictionary	被结束的投票的信息
statsList	NSArray	选项信息
isShow	NSNumber	是否公开结果

statsList说明：

参数	类型	描述
op	NSString	选项名
opNum	NSNumber	选择数
percent	NSNumber	百分比

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOTE_PUB callback:^(id obj) {
//返回数据
{
    info =    {
        bid = 12526;
        endTime = "2016-03-04 18:41:06";
        label = 1;
        nickname = "\U5146\U80fd";
        noticeTime = "18:40";
        optional = 1;
        owner = 1;
        startTime = "03-04 18:40";
        status = 2;
        title = 1;
        uid = "zb_12526";
        vid = 12649;
    };
    isShow = 1;
    statsList =    (
        {
            op = 213;
            opNum = 0;
            percent = 0;
        },
        {
            op = 5646;
            opNum = 0;
            percent = 0;
        },
        {
            op = 879;
            opNum = 0;
            percent = 0;
        }
    );
}
}];
```

获取全部投票

调用示例：

```
[self.talkfunSDK getVotes:^(id result) {

    if ([result isKindOfClass:[NSArray class]]) {
        //弹第0个投票
        dispatch_async(dispatch_get_main_queue(), ^{
            NSMutableArray *array = result;
            if (array.count>0) {
                [weakSelf voteNew: result[0]];
            }

        });
    }

}];
```

获取未收到的投票

调用示例：

```
[self.talkfunSDK getVotesUnreceived:^(id result) {

    if ([result isKindOfClass:[NSArray class]]) {

        dispatch_async(dispatch_get_main_queue(), ^{
            NSMutableArray *array = result;

            if (array.count>0) {
                [weakSelf voteNew: result[0]];
            }

        });
    }

}];
```

抽奖开始

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onLotteryStart:(TFLotteryStart*_Nonnull)result
```

- TFLotteryStart

```
@interface TFLotteryStart : NSObject
/**抽奖编号*/
@property (nonatomic, copy, nonnull)NSString *t;
/**发起抽奖人资料*/
@property (nonatomic, strong, nonnull)TFLotteryStartUser* user;
@end

@interface TFLotteryWinner : NSObject
/**中奖人的编号*/
@property (nonatomic, copy, nonnull)NSString* xid;
/**中奖人名字*/
@property (nonatomic, copy, nonnull)NSString* nickname;
/**发起抽奖人的名字*/
@property (nonatomic, copy, nonnull)NSString* launch_nickname;
/** 是不是自己*/
@property (nonatomic, copy, nonnull) NSString* isMe;
@end
```

抽奖结束

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onLotteryStop:(TFLotteryStop*_Nonnull)result
```

- TFLotteryStop

```
@interface TFLotteryStop : NSObject
@property (nonatomic, copy, nonnull)NSString* gid;
/**是否显示抽奖结果*/
@property (nonatomic, assign)BOOL pub;
@property (nonatomic, copy, nonnull)NSString* lotery_id;
//中奖人列表
@property (nonatomic, strong)NSMutableArray <TFLotteryWinner *>* _Nullable winnerList;
/**奖品名称*/
@property (nonatomic, copy, nonnull)NSString*prize_name;
@end
```

-----下面的代码为过期的方法,不建议使用-----

抽奖开始

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onLotteryStart:(TFLotteryStart*_Nonnull)result
```

TFLotteryStart参数说明：

参数	类型	描述
t	NSString	抽奖编号
user	TFLotteryStartUser	发起抽奖人资料

抽奖结束

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onLotteryStop:(TFLotteryStop*_Nonnull)result
```

TFLotteryStop参数说明：

参数	类型	描述
gid	NSString	gid
pub	BOOL	是否显示抽奖结果
lotery_id	NSString	lotery_id
winnerList	NSMutableArray	中奖人列表
prize_name	NSString	奖品名称

-----下面的代码为过期的方法,不建议使用-----

抽奖

抽奖开始: **TALKFUN_EVENT_LOTTERY_START**

@参数说明：

参数	类型	描述
user	NSDictionary	发起人信息(有发起人名字)

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_LOTTERY_START callback:^(id obj) {
//返回数据
{
    user =    {
        nickname = "\U5146\U80fd";
    };
}
}];
```

抽奖结束： TALKFUN_EVENT_LOTTERY_STOP

@参数说明：

参数	类型	描述
result	NSArray	中奖人信息

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_LOTTERY_STOP callback:^(id obj) {
//返回数据
{
    result =    (
        {
            "launch_nickname" = "\U5146\U80fd";
            "launch_xid" = 300218;
            liveid = 1269050;
            nickname = rrr;
            roomid = 550481;
            time = "03/04 18:43";
            xid = 346440;
        }
    );
}
}];
```

result说明：

参数	类型	描述
xid	NSString	中奖人ID
launch_xid	NSNumber	发起人ID
roomid	NSNumber	房间ID
time	NSString	抽奖结束时间
liveid	NSString	直播ID
nickname	NSString	中奖人名字
launch_nickname	NSString	发起人名字

大班抽奖开始:

```
- (void)onLotteryStart:(TFLotteryStartResult*_Nonnull)result
```

@参数说明:

参数	类型	描述
result	NSDictionary	发起人的信息

result说明:

参数	类型	描述
t	NSString	抽奖编号
user	NSDictionary	发起抽奖人资料

user说明:

参数	类型	描述
gid	NSString	发起抽奖人所在的组
nickname	NSString	名字
role	NSString	角色

大班抽奖结束:

```
- (void)onLotteryStop:(TFLotteryStopResult*_Nonnull)result
```

@参数说明:

result说明:

参数	类型	描述
gid	NSString	发起抽奖人所在的组
pub	BOOL	显示抽奖结果
lotery_id	NSString	中奖ID
winnerList	NSArray	中奖人列表
prize_name	NSString	奖品名称

winnerList里面数据说明:

参数	类型	描述
xid	NSString	中奖人的编号
nickname	NSString	中奖人名字
launch_nickname	NSArray	发起抽奖人的名字
isMe	NSString	是不是自己

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:45:00

初始化时的花朵数:

- flowerTotal

```
self.talkfunSDK.roomInfo.flowerTotal
```

花朵总数更新

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

调用示例:

```
- (void)onFlowerTotal:(NSInteger)total
```

接收别人发送的花朵

- TalkfunSDKLiveDelegate 或 TalkfunSDKLifeLiveDelegate

```
- (void)onFlowerSend:(TFLiveFlowerData*)result
```

发送花朵

- - (void)sendFlower:(void (^__nullable) (NSInteger code, TFLiveFlowerData* _Nullable model, NSInteger flower_left_time))callback

调用示例:

```
[self.talkfunSDK sendFlower:^(NSInteger code, TFLiveFlowerData * _Nullable model, NSInteger flower_left_time) {
    if (code == 0) {
    }else{
    }
}];
```

-----下面的代码为过期的方法,不建议使用-----

花朵初始化: TALKFUN_EVENT_FLOWER_GET_INIT

@参数说明:

参数	类型	描述
code	NSNumber	状态码
amount	NSNumber	花朵数目

调用示例：

```
[self.talkfunSDK on: TALKFUN_EVENT_FLOWER_GET_INIT callback:^(id obj) {
//返回数据

{
amount = 3;
code = 0;
}
}];
```

花朵总数: TALKFUN_EVENT_FLOWER_TOTAL

@参数说明：

参数	类型	描述
total	NSNumber	花朵总数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_FLOWER_TOTAL callback:^(id obj) {
//返回数据
{"total":1}
}];
```

送花: TALKFUN_EVENT_FLOWER_SEND

@参数说明：

参数	类型	描述
amount	NSNumber	花朵总数
nickname	NSString	用户名
role	NSString	用户角色
sendtime	NSNumber	送花时间
time	NSString	现在的时间
uid	NSString	合作方用户ID
xid	NSNumber	用户ID

调用示例：

```
[self.talkfunSDK on: TALKFUN_EVENT_FLOWER_SEND callback:^(id obj) {  
    //返回数据  
    {  
        amount = 3;  
        nickname = rrrr;  
        role = user;  
        sendtime = 1457087230;  
        time = "18:27";  
        uid = 1185252;  
        xid = 337861;  
    }  
}];
```

花朵获取剩余时间： TALKFUN_EVENT_FLOWER_TIME_LEFT

@参数说明：

参数	类型	描述
code	NSNumber	状态码
leftTime	NSNumber	获取花朵的剩余时间

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_FLOWER_TIME_LEFT callback:^(id obj) {  
    //返回数据  
    {  
        code = 15000;  
        leftTime = 173;  
    }  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:05

推送公告的事件回调

- TalkfunSDKLiveDelegate

```
- (void)onAnnouncement:(TFAnnouncement*_Nullable)result
```

接收滚动通知事件回调

- TalkfunSDKLiveDelegate

```
- (void)onAnnounceRoll:(TFAnnounceRoll*_Nullable)result
```

-----下面的代码为过期的方法,不建议使用-----

公告

发布公告: **TALKFUN_EVENT_ANNOUNCE_NOTICE**

@参数说明:

参数	类型	描述
content	NSString	公告内容
time	NSString	公告发布时间

调用示例:

```
[self.talkfunSDK on:TALKFUN_EVENT_ANNOUNCE_NOTICE callback:^(id obj) {  
//返回数据  
{  
"content":"123",  
"time":"2015-12-17 16:01"  
}  
}];
```

滚动通知: **TALKFUN_EVENT_ANNOUNCE_ROLL**

@参数说明:

参数	类型	描述
content	NSString	内容
link	NSString	链接地址
time	NSNumber	时间戳

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_ANNOUNCE_ROLL callback:^(id obj) {  
//返回数据  
{  
    content = 45312;  
    duration = 600;  
    link = "http://baidu.com";  
    time = 1457088360;  
}  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:44:59

接收广播

- TalkfunSDKLiveDelegate

```
- (void)onBroadcast:(TFBroadcast*)resul
```

获取全部的广播

```
-(void)getBroadcastsList:(void (^__nullable)(NSInteger code ,NSMutableArray <TFBroadcast
*>*__Nullable list ,TFError *__Nullable error))callback
```

获取未收到的广播

```
-(void)getBroadcastsUnreceivedList:(void (^__nullable)(NSInteger code ,NSMutableArray <TF
Broadcast *>*__Nullable list ,TFError *__Nullable error))callback
```

-----下面的代码为过期的方法,不建议使用-----

广播

广播: TALKFUN_EVENT_BROADCAST

@参数说明:

参数	类型	描述
message	NSString	广播信息
__auto	NSNumber	是否自动发给后来者
uniqid	NSString	广播唯一标识
t	NSNumber	时间戳

调用示例:

```
[self.talkfunSDK on:TALKFUN_EVENT_BROADCAST callback:^(id obj) {
//返回数据
{
    "__auto" = 1;
    message = "854321.0";
    t = 1457086318;
    uniqid = 56d95f6e08367;
}
}];
```

获取全部的广播

调用示例：

```
[self.talkfunSDK getBroadcasts:^(id result) {

    if ([result isKindOfClass:[NSArray class]]) {
        dispatch_async(dispatch_get_main_queue(), ^{
            NSMutableArray *array = result;

            if (array.count>0) {
                [weakSelf broadcast: result[0]];
            }

        });
    }

}];
```

获取未收到的广播

调用示例：

```
[self.talkfunSDK getBroadcastsUnreceived:^(id result) {

    if ([result isKindOfClass:[NSArray class]]) {
        dispatch_async(dispatch_get_main_queue(), ^{
            NSMutableArray *array = result;

            if (array.count>0) {
                [weakSelf broadcast: result[0]];
            }

        });
    }

}];
```

获取网络线路组

- - (void)getLineList:(void (^)(_Nullable)(TFLiveLineManager * _Nullable result))callback;

调用示例：

```
[self.talkfunSDK getLineList:^(TFLiveLineManager* result) {
    if (result.code == TalkfunCodeSuccess) {

    }else{

    }

}];
```

设置线路

- - (void)setNetwork:(NSString _Nonnull)operatorID selectedSegmentIndex:(NSInteger)selectedSegmentIndex callback:(void (^)(_nullable)(NSInteger code ,TFError __Nullable error))callback

调用示例：

```
[self.talkfunSDK setNetwork:button.networkDict.key selectedSegmentIndex:button.groupIndex callback:^(NSInteger code, TFError * _Nullable error) {

    if (code == 0) {

    }

}];
```

-----下面的代码为过期的方法,不建议使用-----

伪直播说明

- 伪直播和直播的播放流程和逻辑是一致的，不需要特殊处理。
- 网络线路只能切换不同线路不能选择运营商
- 如果当前网络速度不理想，SDK提供了接口更改网络。

网络选择

```
//获取网络列表(伪直播和直播的播放流程和逻辑是一致的)
- (void)getNetworkList:(void (^)(id result))callback;

//设置网络
- (void)setNetwork:(NSString *)operatorID sourceName:(NSString *)sourceName selectedSegmentIndex:(NSInteger)selectedSegmentIndex;
```

调用示例：

```
weakSelf

// isVodLive为YES,是伪直播

[self.talkfunSDK getNetworkList:^(id result) {

    if ([result[@"code"] intValue] == TalkfunCodeSuccess) {

        //判断如果是直播
        if ([result[@"isVodLive"] intValue ]==0)
        {

            weakSelf.networkSelectionVC.networkSelectionArray = result[@"data"];

            PERFORM_IN_MAIN_QUEUE(
                //TODO:网络选择控制器
                [weakSelf.view addSubview:weakSelf.networkSelectionVC.view];
            )

        }

        //判断如果是伪直播
        else{

            UIView * view = [[UIApplication sharedApplication].keyWindow viewWithTag:123];

            if (!view) {

                //TODO:线路选择
                TalkfunNetworkLines * networkLinesView = [[NSBundle mainBundle] loadNibNamed:@"TalkfunNetworkLines" owner:nil options:nil][0];
                networkLinesView.tag = 123;
            }
        }
    }
}];
```

```

        networkLinesView.networkLinesArray = result[@"data"][@"operators"];

        networkLinesView.frame = CGRectMake(0, 0, ScreenSize.width, ScreenSize.height);

        weakSelf
        networkLinesView.networkLineBlock = ^(NSNumber * networkLineIndex){

            [weakSelf.talkfunSDK setNetwork:nil selectedSegmentIndex:[networkLineIndex intValue]];

        };
        PERFORM_IN_MAIN_QUEUE(
            [weakSelf.view addSubview:networkLinesView];
        )
    }

}

}else{
    [weakSelf.view toast:result[@"msg"] position:ToastPosition];
}

}];

```

@直播参数说明

参数	类型	描述
code	NSNumber	状态码
data	NSArray	数据
network	NSDictionary	网络信息
cdn	NSString	cdn

直播data数据说明:

参数	类型	描述
cur	NSNumber	线路
sourceName	NSString	资源名称
host	NSString	域名
cdn	NSString	cdn

@伪直播参数说明: 数组里面装的是字典，字典里面数据如下：

参数	类型	描述
current	NSNumber	是否当前线路（0不是，1是）
name	NSString	线路名
index	NSNumber	线路指针（调用setNetworkLine传进的参数）

把对应的运营商的key作为operatorID参数、sourceName作为sourceName参数、cur作为cur参数传进去。

直播

```
-(void)setNetwork:(NSString *)operatorID sourceName:(NSString *)sourceName selectedSegmentIndex:(NSInteger)selectedSegmentIndex;
```

伪直播

```
[self.talkfunSDK setNetwork:@"operatorID" sourceName:@"sourceName" selectedSegmentIndex:@"cur"];
```

网络速度: TALKFUN_EVENT_NETWORK_SPEED

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_NETWORK_SPEED callback:^(id obj) {
}];
```

@参数说明：

参数	类型	描述
speed	NSString	当前速度
networkStatus	TalkfunNetworkStatus	网络状态

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 18:38:37

pragma mark 音视频代理 TalkfunSDKLiveDelegate

@参数说明: //0:应用,1:播放,2:暂停,3:拖动,4:关闭//多媒体播放状态(音视频) typedef enum {

```
TalkfunMultiMediaStatusChangeListenerPlaying      = 1, //播放
TalkfunMultiMediaStatusChangeListenerStop        = 2, //停止
TalkfunMultiMediaStatusChangeListenerSeek        = 3, //快进
TalkfunMultiMediaStatusChangeListenerClose       = 4, //关闭
TalkfunMultiMediaStatusChangeListenerComplate    = 5, //完成
TalkfunMultiMediaStatusChangeListenerError       = 6, //错误
```

```
}TalkfunMultiMediaStatusChangeListener;
```

```
/**
 * @param status 播放状态
 * @param time   播放时间
 */
- (void)multiMediaApplicate:(TalkfunMultiMediaStatusChangeListener)status time:(int)time
;
```

```
/* 应用
 * @param id 素材id
 * @param docType 5:音频 ,4:视频
 * @param title 素材标题
 * @param duration 视频时长
 */
-(void)multiMediaApplicatea:(NSString*)ID docType:(int)type title:(NSString*)title durati
on:(int)duration
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间: 2020-05-13 15:42:51

extern NSString * const TALKFUN_EVENT_RTC_OPEN; 开启讲台模式

```
//TODO:开启讲台模式
[self.talkfunSDK on:TALKFUN_EVENT_RTC_OPEN callback:^(id obj) {

}};
```

extern NSString * const TALKFUN_EVENT_RTC_CLOSE; 关闭讲台模式

```
//TODO:老师关闭了讲台模式
[self.talkfunSDK on:TALKFUN_EVENT_RTC_CLOSE callback:^(id obj) {

}};
```

extern NSString * const TALKFUN_EVENT_RTC_LIVE_DURATION; 直播时长

参数	类型	描述
year	NSString	年
month	NSString	月
day	NSString	日
hour	NSString	小时
minute	NSString	分钟
second	NSString	秒

```
//TODO:直播时长
[self.talkfunSDK on:TALKFUN_EVENT_RTC_LIVE_DURATION callback:^(id obj) {

}};
```

extern NSString * const TALKFUN_EVENT_RTC_AWARD; 奖励

```
[self.talkfunSDK on: TALKFUN_EVENT_RTC_AWARD callback:^(id obj) {

}};
```

extern NSString * const TALKFUN_EVENT_RTC_CONNECTIONINTERRUPTED; 频道连接断开


```
//TODO:SDK与频道断开连接
[self.talkfunSDK on:TALKFUN_EVENT_RTC_CONNECTIONINTERRUPTED callback:^(id obj) {

}};
```

extern NSString * const TALKFUN_EVENT_RTC_RE_CNNECT; 重新加入频道成功

```
//SDK重新加入频道
[self.talkfunSDK on:TALKFUN_EVENT_RTC_RE_CNNECT callback:^(id obj) {

}};
```

-----下面所有 允许上讲台 的主要参数说明-----

参数	类型	描述
view	UIView	视频预览容器
dict	用户数据	字典

----- 允许上讲台内dict主要参数说明-----

参数	类型	描述
audio	NSString	音频权限
drawPower	NSString	涂鸦权限
role	NSString	角色(有老师与同学)
time	NSString	送花
video	NSString	视频权限
xid	NSString	唯一标识
isMe	NSString	是不是是自己

extern NSString * const TALKFUN_EVENT_RTC_USER_UP; 允许上讲台

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_USER_UP callback:^(id obj) {

}};
```

```

typedef NS_ENUM(NSInteger, TalkfunApplyStatus) {
    // 未申请 NoApply
    TalkfunApplyStatusNoApply = 0,
    // 申请上讲台
    TalkfunApplyStatusApplying = 1,
    // 讲台上
    TalkfunApplyStatusAllow = 2,
};

typedef NS_ENUM(NSInteger, TalkfunMediaPower) {
    //老师操作关闭设备
    TalkfunMediaPowerCloseForZHUBO = 0,
    //老师操作打开设备
    TalkfunMediaPowerOpenForZHUBO = 1,
    //学员操作关闭设备
    TalkfunMediaPowerCloseForUser = 10,
    //学员操作打开设备
    TalkfunMediaPowerOpenForUser = 11,
};

@interface TalkfunSDKPlatform : NSObject

//申请状态 ApplyStatus
@property (nonatomic, assign) TalkfunApplyStatus applyStatus;

```

-----下面所有 **result** 的主要参数说明-----

参数	类型	描述
audio	NSString	音频权限
drawPower	NSString	涂鸦权限
role	NSString	角色(有老师与同学)
time	NSString	送花
video	NSString	视频权限
xid	NSString	唯一标识
isMe	NSString	是不是是自己

extern NSString * const TALKFUN_EVENT_RTC_KICK; 踢下讲台

```

[self.talkfunSDK on:TALKFUN_EVENT_RTC_KICK callback:^(id result) {

}];

```

extern NSString * const TALKFUN_EVENT_RTC_DOWN; 有人主动下讲台

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_DOWN callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_OPEN_VIDEO; 打开摄像头

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_OPEN_VIDEO callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_CLOSE_VIDEO; 关闭摄像头

```
[self.talkfunSDK on: TALKFUN_EVENT_RTC_CLOSE_VIDEO callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_OPEN_AUDIO; 打开麦克风

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_OPEN_AUDIO callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_CLOSE_AUDIO; 关闭麦克风

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_CLOSE_AUDIO callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_DRAW_ENABLE; 打开画线权限

```
[self.talkfunSDK on: TALKFUN_EVENT_RTC_DRAW_ENABLE callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_DRAW_DISABLE; 关闭画线权限

```
[self.talkfunSDK on: TALKFUN_EVENT_RTC_DRAW_DISABLE callback:^(id result) {  
  
}];
```

extern NSString * const TALKFUN_EVENT_RTC_OFFLINE ; 用户离线

```
[self.talkfunSDK on:TALKFUN_EVENT_RTC_OFFLINE callback:^(id result) {  
  
}];
```

申请上讲台

- (void)apply:(void (^)(NSDictionary* result))callback;

```
[weakSelf.talkfunSDK.platformManager apply:^(NSDictionary* result) {  
  
}];
```

取消上讲台申请

- (void)cancel:(void (^)(NSDictionary* result))callback;

```
[weakSelf.talkfunSDK.platformManager cancel:^(NSDictionary* result) {  
  
}];
```

主动下讲台

- (void)down:(void (^)(NSDictionary* result))callback;

```
[weakSelf.talkfunSDK.platformManager down:^(NSDictionary* result) {  
  
}];
```

打开摄像头

- (void)openVideo:(void (^)(NSDictionary* result))callback;

```
[self.talkfunSDK.platformManager openVideo:^(NSDictionary *result) {  
  
  
}];
```

关闭摄像头

- (void)closeVideo:(void (^)(NSDictionary* result))callback;

```
[self.talkfunSDK.platformManager closeVideo:^(NSDictionary *result)
    ]];
```

打开麦克风

- (void)openAudio:(void (^)(NSDictionary* result))callback;

```
[self.talkfunSDK.platformManager openAudio:^(id result) {
}];
```

关闭麦克风

- (void)closeAudio:(void (^)(NSDictionary* result))callback;

```
[self.talkfunSDK.platformManager closeAudio:^(id result) {
}];
```

...

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

申请连麦

调用示例：

```
[self.talkfunSDK.rtcManager applyRTC:^(NSInteger code, TFError * _Nullable error) {

    if (code == 0) {

    }else {

    }

}];
```

取消申请连麦

调用示例：

```
[self.talkfunSDK.rtcManager cancelRTC:^(NSInteger code, TFError * _Nullable error) {

    if (code == 0) {

    }else {

    }

}];
```

同意被邀请连麦

调用示例：

```
[self.talkfunSDK.rtcManager respondinviteRTC:TalkfunRespondinviteStatusAccept callback:^(
NSInteger code, TFError * _Nullable error) {

    if(code==0){

    }else{
        [weakSelf toast:error.msg?error.msg:@"拒绝上讲台"];
    }

}];
```

拒绝被邀请连麦

调用示例：

```
[self.talkfunSDK.rtcManager respondinviteRTC:TalkfunRespondinviteStatusReject callback:^(
NSInteger code, NSError * _Nullable error) {

    if(code==0){

    }else{
        [weakSelf toast:error.msg?error.msg:@"拒绝上讲台"];
    }

}];
```

-----TalkfunSDKRtcDelegate-----

讲台开启

调用示例：

```
- (void)rtcOpen
{
}
```

讲台关闭

调用示例：

```
- (void)rtcClose
{
}
```

被邀请连麦

调用示例：

```
- (void)rtcInvite
{
}
```

被取消邀请连麦

调用示例：

```
- (void)rtcInvitecancel
{
}
```

主播拒绝学员的连麦申请

调用示例：

```
- (void)rtcRejectapply  
{  
}
```

允许连麦

调用示例：

```
- (void)rtcUp:(TalkfunRtcModel*)model  
{  
}
```

学员主动取消连麦

调用示例：

```
- (void)rtcDown:(TalkfunRtcModel*)model  
{  
}
```

取消连麦 (主播踢的)

调用示例：

```
- (void)rtcKick:(TalkfunRtcModel*)model  
{  
}
```

用户离线

调用示例：

```
- (void)rtcOffline:(TalkfunRtcModel*)model  
{  
}
```

打开摄像头

调用示例：

```
- (void)rtcOpenVideo:(TalkfunRtcModel*)model  
{  
}
```

关闭摄像头

调用示例：

```
- (void)rtcCloseVideo:(TalkfunRtcModel*)model
{
}
```

打开麦克风

调用示例：

```
- (void)rtcOpenAudio:(TalkfunRtcModel*)model
{
}
```

关闭麦克风

调用示例：

```
- (void)rtcCloseAudio:(TalkfunRtcModel*)model
{
}
```

奖励

调用示例：

```
- (void)rtcAward:(TalkfunRtcModel*)model
{
}
```

奖励

//小班全体互动事件 // 全体音频打开 // 全体音频关闭 // 全体视频打开 // 全体视频关闭 // 全体音视频打开 // 全体音视频关闭

调用示例：

```
- (void)onGlobalStatus:(TalkfunGlobalStatus)status
{
}
```

TalkfunRtcModel参数说明：

参数	类型	描述
role	int	角色
xid	NSString	用户id
score	NSString	奖励
time	NSString	时间戳
nickname	NSString	名字
drawPower	BOOL	涂鸦权限
isMe	BOOL	奖励
video	int	用户的视频开关权限
audio	int	用户的音频开关权限
view	UIView	用户视频预览容器
uid	NSString	UID
avatar	NSString	头像
voice_counts	NSString	声音音量大小
isShowNickName	BOOL	显示昵称
isLoading	NSString	显示断网动画
isFull	BOOL	是否全屏

-----下面的代码为过期的方法,不建议使用-----

被邀请连麦

- 监听 TALKFUN_EVENT_RTC_INVITE

```

block调用示例：
[self.talkfunSDK on: TALKFUN_EVENT_RTC_INVITE callback:^(id obj) {
//返回数据
{

}
}
}];

```

同意被邀请连麦

block调用示例：

```
[self.talkfunSDK respondinvite:TalkfunRespondinviteStatusAccept callback:^(id result) {  
//返回数据  
{  
"code":"0"  
  
}  
}];
```

拒绝被邀请连麦

block调用示例：

```
[self.talkfunSDK respondinvite:TalkfunRespondinviteStatusReject callback:^(id result) {  
//返回数据  
{  
"code":"0"  
  
}  
}];
```

取消邀请连麦

- 监听 TALKFUN_EVENT_RTC_INVITECANCEL

block的调用示例：

```
[self.talkfunSDK on: TALKFUN_EVENT_RTC_INVITECANCEL callback:^(id obj) {  
//返回数据  
{  
  
}  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:05

TalkfunSDKLifeLiveDelegate

房间事件回调

- 如果加入成功，code 会是一个正数（code > 0），代表加入房间成功）。error 为null
- 如果加入失败，code 会是一个负数（code < 0），代表进房失败的错误码。error 有值
- 调用刷新refreshData: 也会重新回调
- 进房失败的错误码含义请参见[错误码 TalkfunSDK 文件的TalkfunError 枚举

```
- (void)onEnterRoom:(NSInteger)code result:(TFError*_Nonnull)error
```

接收别人的聊天事件回调

- 详见 TFChatSend.h 中的定义

```
- (void)onDidReceiveText:(TFChatData*_Nullable)result
```

当前用户进入房间触发回调

- 详见 TFMember.h 中的定义

```
- (void)onMemberJoinMe:(TFMember*_Nullable)result
```

其它人进入房间触发回调

- 详见 TFMember.h 中的定义

```
- (void)onMemberJoinOther:(TFMember*_Nullable)result
```

有人退出房间

- 详见 TFMember.h 中的定义

```
- (void)onMemberLeave:(TFMember*_Nullable)result
```

同时在线人数 (真实人+虚拟人员 的总数)

```
- (void)onPeakConcurrentUsers:(NSInteger)total
```

摄像头开启

```
- (void)onCameraShow
```

摄像头关闭

```
- (void)onCameraHide
```

直播间状态

```
- (void)onLiveStatus:(TalkfunLiveStatus)status
```

接收滚动通知事件回调

- 详见 TFAnnounceRoll.h 中的定义

```
- (void)onAnnounceRoll:(TFAnnounceRoll*_Nullable)result
```

当前用户被踢出房间的事件回调

```
- (void)onMemberKick
```

推送弹窗广告的事件回调

```
- (void)onPopupPut:(TFPopupPut*_Nullable)result
```

商品列表的事件回调

- 商品状态变更时会实时更新列表
- self.lifeConfig.store.goods 与当前goods 是同一个列表

```
- (void)onGoodsUpdate:(NSMutableArray <TFgoods *>* _Nullable)goodsList
```

推送公告的事件回调

```
- (void)onAnnouncement:(TFAnnouncement*_Nullable)result
```

接收主播发起签到的事件回调

```
- (void)onSignNew:(TFSignNew*_Nullable)result
```

接收主播结束签到的事件回调

```
- (void)onSignEnd:(TFSignEnd*_Nullable)result
```

投票开始的事件回调

```
- (void)onVoteNew:(TFVoteData*_Nullable)result
```

投票结束的事件回调

```
- (void)onVotePub:(TFVoteData*_Nullable)result
```

删除投票的事件回调

```
- (void)onVoteDel:(TFVoteData*_Nullable)result
```

接收别人私聊的回调

```
- (void)onChatPrivate:(TFChatData*_Nullable)result
```

在线客服列表变更时会实时更新列表

```
- (void)onCustomerServiceListUpdate:(NSMutableArray <TFMember *>* _Nullable)list
```

主播回答了问题

```
- (void)onQuestionReply:(TFQuestionData*_Nonnull)result
```

提问删除

```
- (void)onQuestionDelete:(TFQuestionData*_Nonnull)result
```

接收别人的提问

```
- (void)onQuestionAsk:(TFQuestionData*_Nonnull)result
```

提问审核通过

```
- (void)onQuestionAudit:(TFQuestionData*_Nonnull)result
```

抽奖开始

```
- (void)onLotteryStart:(TFLotteryStart*_Nonnull)result
```

抽奖结束

```
- (void)onLotteryStop:(TFLotteryStop*_Nonnull)result
```

单体禁言

- 单体禁言 :YES
- 单体解除禁言 :NO

```
- (void)onChatDisable:(BOOL)state
```

全体禁言

- 全体禁言:YES
- 全体解除禁言:NO

```
- (void)onChatDisableAll:(BOOL)state;
```

监听别人的点赞

```
- (void)onLikePut:(TFLikeData*_Nullable)likeData
```

-----属性与方法-----

代理对象

```
@property (nonatomic, weak) id<TalkfunSDKLifeLiveDelegate> _Nullable delegate;
```

配置视频窗口容器

```
@property (nonatomic, strong) UIView * _Nullable cameraContainerView;
```

音量

```
@property (nonatomic, assign) CGFloat playVolume;
```

是否显示访客人数

- onEnterRoom:result:()回调后,visitors的值才准确

```
@property (nonatomic, assign) BOOL visitors;
```

主播数据

- onEnterRoom:result:()回调后,zhubo的值才准确
- 详见 TFZhubo.h 中的定义

```
@property (nonatomic, strong) TFZhubo * _Nullable zhubo;
```

摄像头开关

- onEnterRoom:result:()回调后, isCamera的值才准确

```
@property (nonatomic, assign) BOOL isCamera;
```

单体禁言

- onEnterRoom:result:()回调后, isChatDisables的值才准确

```
@property (nonatomic, assign) BOOL isChatDisables;
```

全体禁言

- onEnterRoom:result:()回调后, isDisableall的值才准确

```
@property (nonatomic, assign) BOOL isDisableall;
```

同时在线人数

- onEnterRoom:result:()回调后, peakConcurrentUsers的值才准确

```
@property (nonatomic, assign) NSInteger peakConcurrentUsers;
```

课程开始的时间戳

- onEnterRoom:result:()回调后, startTime的值才准确


```
@property (nonatomic, assign) NSInteger startTime;
```

获取生活直播相关配置

- onEnterRoom:result:()回调后, roomInfo的值才准确

```
@property (nonatomic, strong) TFLifeRoomInfo * _Nullable roomInfo;
```

课程开始的时间戳

- onEnterRoom:result:()回调后, liveStatus的值才准确

直播间状态

```
@property (nonatomic, assign) TalkfunLiveStatus liveStatus;
```

默认点赞数

- onEnterRoom:result:()回调后, LikeResult的值才准确

```
@property (nonatomic, strong) TFLikeData * _Nullable likeResult;
```

在线的客服列表

- onEnterRoom:result:()回调后, serviceList的值才准确

```
@property (nonatomic, strong) NSMutableArray<TFMember *> _Nullable serviceList;
```

视频显示模式

- 默认 TalkfunMovieScalingModeAspectFit

```
@property (nonatomic, assign) TalkfunMovieScalingMode scalingMode;
```

创建 TalkfunSDKLifeLive 单例

```
+ (instancetype _Nullable ) sharedInstance;
```

销毁 TalkfunSDKLifeLive 单例

```
- (void) destroySharedIntance;
```

进入房间

- 调用 enterRoom() 接口执行进房操作后，会收到来自 SDK 的 onEnterRoom:result() 回调：
- @param param 进房参数，请参考 TFINitParams

```
- (void)enterRoom:(TFInitParams *_Nullable)param;
```

发送聊天

- result result = 0 成功)，result != 0 时为错误码。

```
- (void)sendChat:(NSString *_Nullable)msg callback:(void (^__nullable) (NSInteger code, TFChatData *_Nullable model, TFError *_Nullable error))callback;
```

发送私聊客服

- result result = 0 成功)，result != 0 时为错误码。

```
- (void)sendChatPrivate:(NSString *_Nullable)xid msg:(NSString *_Nullable)msg callback:(void (^__nullable)(NSInteger code, TFChatData *_Nullable model, TFError *_Nullable error))callback;
```

发起提问

```
- (void)enterAskQuestions:(NSString *_Nullable)msg callback:(void (^__nullable) (NSInteger code, TFQuestionData *_Nullable model, TFError *_Nullable error))callback;
```

获取提问列表

```
- (void)getAListOfQuestions:(void (^__nullable) (NSInteger code, NSMutableArray <TFQuestionData * > *_Nullable list, TFError *_Nullable error))callback;
```

刷新SDK

```
- (void)refreshData;
```

获取网络线路组 2.0

```
- (void)getLineList:(void (^__Nullable)(TFLiveLineManager *_Nullable result))callback;
```

设置线路 2.0

- 例子
- "operatorID" : "telecom" 网络类型
- "selectedSegmentIndex" : 0, 组索引

```
- (void)setNetwork:(NSString *_Nonnull)operatorID selectedSegmentIndex:(NSInteger)selectedSegmentIndex callback:(void (^__nullable)(NSInteger code ,TFError *_Nullable error))callback;
```

点赞

```
- (void)likeClick:(void (^__nullable)(NSInteger code , TFLikeData *_Nullable model,TFError *_Nullable error))callback;
```

获取邀请榜单

```
- (void)getInvitationList:(void (^__nullable)(NSInteger code , TFInvitationListData *_Nullable model,TFError *_Nullable error))callback;
```

举报主播

```
- (void)reportingMembers:(NSString*_Nullable)msg callback:(void (^__nullable)(NSInteger code , TFError *_Nullable error))callback;
```

配置ppt容器（可用于重新设置ppt容器）

- 初始化完SDK后即可调用

```
- (void)configurePPTContainerView:(UIView *_Nonnull)pptContainerView;
```

进入后台时是否暂停音频(默认暂停YES)

```
- (void)setPauseInBackground:(BOOL)pause;
```

签到

```
- (void)signIn:(NSString*_Nullable)signId callback:(void (^__nullable)(NSInteger code , TFError *_Nullable error))callback;
```

获取投票列表 2.0

```
- (void)getVotingList:(void (^__nullable)(NSInteger code ,NSMutableArray <TFVoteData *>* _Nullable list ,TFError *_Nullable error))callback;
```

获取投票ID详情 2.0

```
- (void)getVotingDetails:(NSString*_Nullable)vid callback:(void (^__nullable)(NSInteger code , TFVoteData *_Nullable model,TFError *_Nullable error))callback;
```

确认投票 2.0

```
- (void)voteSubmit:(TFSubmit*_Nullable)model callback:(void (^__nullable)(NSInteger code , TFError *_Nullable error))callback;
```

上传图片生成url链接

- 如果发送成功，code 会是0 error 等于null imageUrl 会有值
- 如果发送失败，code 会是!=0 error会有值

```
- (void)uploadingpicImg:(UIImage*_Nonnull)image callback:(void (^__nullable)(NSInteger code , NSString *_Nullable imageUrl,TFError *_Nullable error))callback;
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间： 2021-05-25 15:52:42

发送点赞

```
- (void)likeClick:(void (^__nullable)(NSInteger code , TFLikeData *_Nullable model,TFError *_Nullable error))callback;
```

默认点赞数据

- onEnterRoom:result()回调后, LikeResult的值才准确

```
@property (nonatomic, strong)TFLikeData * _Nullable likeResult;
```

监听别人的点赞

- TalkfunSDKLifeLiveDelegate

```
- (void)onLikePut:(TFLikeData*_Nullable)likeData
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:05

商品列表

- onEnterRoom:result:()回调后, goods的值才准确

商品开关

```
self.livingSDK.roomInfo.pageConfig.store.enable
```

商品列表

```
NSMutableArray *goods = self.livingSDK.roomInfo.pageConfig.store.goods
```

商城链接

- onEnterRoom:result:()回调后, enable的值才准确

商城链接

```
self.livingSDK.roomInfo.pageConfig.menu.store.enable
```

```
NSString *url = self.livingSDK.roomInfo.pageConfig.menu.store.url; NSLog(@"内宝贝链接%@", url);
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间 : 2021-05-25 18:38:37

商品列表

- onEnterRoom:result()回调后, roomInfo的值才准确

```
self.livingSDK.roomInfo.pageConfig.store.goods
```

```
@interface TFgoods: NSObject
//状态码
typedef enum{

    TalkfunPutawayOff = 0, //下架
    TalkfunPutawayOn = 1, //上架
    TalkfunPutawayRecommend = 2 //推荐

}TalkfunPutaway;

///【字段含义】 id
@property (nonatomic, copy, nonnull) NSString* goodsId;
///【字段含义】二维码链接
@property (nonatomic, copy, nonnull) NSString* qrcode;
///【字段含义】商品链接
@property (nonatomic, copy, nonnull) NSString* url;
///【字段含义】商品头像
@property (nonatomic, copy, nonnull) NSString* img;
///【字段含义】商品下方的标题
@property (nonatomic, copy, nonnull) NSString* link_text;
///【字段含义】售价 99
@property (nonatomic, copy, nonnull) NSString* now_Price;
///【字段含义】原价 999
@property (nonatomic, copy, nonnull) NSString* originalPrice;
///【字段含义】标签
// 特价 限时 新品 钜惠 秒杀
// 1 2 3 4 5
@property (nonatomic, copy, nonnull) NSString* tab;
///【字段含义】支付方式
@property (nonatomic, copy, nonnull) NSString* pay;
///【字段含义】名字
@property (nonatomic, copy, nonnull) NSString* name;
///【字段含义】
@property (nonatomic, copy, nonnull) NSString* loading1;
///【字段含义】
@property (nonatomic, copy, nonnull) NSString* loading2;
///【字段含义】1上架 0下架 2推荐
@property (nonatomic, assign) TalkfunPutaway putaway;

@end
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 17:08:30

邀请榜单开关

- onEnterRoom:result:()回调后, enable的值才准确
- enable = 1 开
- enable = 0 关

```
NSInteger enable = self.livingSDK.roomInfo.pageConfig.invitationList.enable;
```

获取邀请榜单

- - (void)getInvitationList:(void (^__nullable)(NSInteger code , TFInvitationListData _Nullable model,TFError _Nullable error))callback

调用示例：

```
[self.talkfunSDK getInvitationList:^(NSInteger code, TFInvitationListData * _Nullable model, TFError * _Nullable error) {  
  
    if (code == 0) {  
        }else{  
  
    }  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 18:38:37

举报主播

- - (void)reportingMembers:(NSString_Nullable)msg callback:(void (^_nullable)(NSInteger code , NSError *_Nullable error))callback

调用示例：

```
[self.talkfunSDK reportingMembers:msg callback:^(NSInteger code, NSError * _Nullable error) {  
    if (code == 0) {  
  
    }else{  
  
    }  
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-05-25 16:11:05

推送弹窗广告

```
@interface TFPopupPut : NSObject

    【字段含义】广告播放时长
    @property (nonatomic, assign)NSInteger duration;

    【字段含义】广告图片的链接
    @property (nonatomic, copy, nonnull) NSString* img;

    【字段含义】广告生成的时间 2021-03-09 15:30:38
    @property (nonatomic, copy, nonnull) NSString* pushTime;

    【字段含义】广告图片点击时会跳转的链接
    @property (nonatomic, copy, nonnull) NSString* url;

    【字段含义】发起推广的用户标记
    @property (nonatomic, copy, nonnull) NSString* uuid;
@end
```

浮动广告位置

```
@interface TFposition : NSObject
    @property (nonatomic, assign)NSInteger top;
    @property (nonatomic, assign)NSInteger left;
@end
```

浮动广告开关

```
@interface TFAd : NSObject
    @property (nonatomic, assign)NSInteger enable;
@end
```

浮动广告模型

```
@interface TFAdModel : NSObject
    【字段含义】名字
    @property (nonatomic, copy, nonnull) NSString* name;
    【字段含义】图片URL
    @property (nonatomic, copy, nonnull) NSString* img;
    【字段含义】图片创建时间
    @property (nonatomic, copy, nonnull) NSString * time;
    【字段含义】link
    @property (nonatomic, copy, nonnull) NSString * link;
    【字段含义】 图片高度
    @property (nonatomic, assign)NSInteger height;
    【字段含义】 图片宽度
    @property (nonatomic, assign)NSInteger width;
    【字段含义】缩放
    @property (nonatomic, assign)NSInteger scale;
    【字段含义】位置
    @property (nonatomic, strong)TFposition* _Nonnull position;
@end
```

商品

```
@interface TFgoods: NSObject
//状态码
typedef enum{

    TalkfunPutawayOff = 0, //下架
    TalkfunPutawayOn  = 1, //上架
    TalkfunPutawayRecommend = 2 //推荐

}TalkfunPutaway;

    【字段含义】 id
@property (nonatomic, copy, nonnull) NSString* goodsId;
    【字段含义】 二维码链接
@property (nonatomic, copy, nonnull) NSString* qrcode;
    【字段含义】 商 品链接
@property (nonatomic, copy, nonnull) NSString* url;
    【字段含义】 商 品头像
@property (nonatomic, copy, nonnull) NSString* img;
    【字段含义】 商 品下方的标题
@property (nonatomic, copy, nonnull) NSString* link_text;
    【字段含义】 售 价 99
@property (nonatomic, copy, nonnull) NSString* now_Price;
    【字段含义】 原 价 999
@property (nonatomic, copy, nonnull) NSString* originalPrice;
    【字段含义】 标签
// 特 价 限 时 新 品 钜 惠 秒 杀
//   1   2   3   4   5
@property (nonatomic, copy, nonnull) NSString* tab;
    【字段含义】 支 付 方 式
@property (nonatomic, copy, nonnull) NSString* pay;
    【字段含义】 名 字
@property (nonatomic, copy, nonnull) NSString* name;
    【字段含义】
@property (nonatomic, copy, nonnull) NSString* loading1;
    【字段含义】
@property (nonatomic, copy, nonnull) NSString* loading2;
    【字段含义】 1上架 0下架 2推荐
@property (nonatomic, assign) TalkfunPutaway putaway;

@end
```

选项卡数据

```

@interface TFtabsData: NSObject
    【字段含义】 标题
@property (nonatomic, copy, nonnull) NSString* title;
    【字段含义】 内容
@property (nonatomic, copy, nonnull) NSString* content;
    【字段含义】 名字
@property (nonatomic, copy, nonnull) NSString* name;
    【字段含义】 类型 chat聊天  invite榜单 editor编辑  courseware课件  ...
@property (nonatomic, copy, nonnull) NSString* type;
@end

```

当前用户数据

```

//@interface TFLifeUserInfo : NSObject
**名字*/
//@property (nonatomic, copy, nonnull) NSString* nickname;
**用户头像*/
//@property (nonatomic, copy, nonnull) NSString* avatar;
** 是不是自己*/
//@property (nonatomic, copy, nonnull) NSString* isMe;
**角色
// extern NSString * const TalkfunMemberRoleSpadmin; //超级管理员(老师)
// extern NSString * const TalkfunMemberRoleAdmin; //管理员(助教)
// extern NSString * const TalkfunMemberRoleUser; //普通用户(学生)
// extern NSString * const TalkfunMemberRoleGuest; //游客
// */
//@property (nonatomic, copy, nonnull) NSString* role;
//@property (nonatomic, copy, nonnull) NSString* gender;
//@property (nonatomic, copy, nonnull) NSString* uid;
//@property (nonatomic, copy, nonnull) NSString* xid;
//@property (nonatomic, copy, nonnull) NSString* partner_id;
//@property (nonatomic, copy, nonnull) NSString* expire;
//@property (nonatomic, copy, nonnull) NSString* pid;
//@property (nonatomic, copy, nonnull) NSString* course_id;
//@property (nonatomic, copy, nonnull) NSString* qaSecond;
//@property (nonatomic, copy, nonnull) NSArray * attr;
//@property (nonatomic, copy, nonnull) NSString * roomid;
//@property (nonatomic, copy, nonnull) NSString * a;
//@property (nonatomic, copy, nonnull) NSString * gid;
//@property (nonatomic, copy, nonnull) NSString * rname;
//@property (nonatomic, copy, nonnull) NSString * sessionid;
//@property (nonatomic, copy, nonnull) NSString* regtime;
//@end

```

商城开关

```

@interface TFStore : NSObject
    【字段含义】是否启用
@property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 ( 默认: " ")
@property (nonatomic, copy, nonnull)NSString * url;
    【字段含义】名字 ( 默认: "二维码链接 ")
@property (nonatomic, copy, nonnull)NSString * qrcode;
    【字段含义】名字 ( 默认: "商_城入口 ")
@property (nonatomic, copy, nonnull)NSString * name;
    【字段含义】名字 ( 默认: 类型 1商品列表 2商品链接 3商品二维码 )
@property (nonatomic, assign)NSInteger type;

//商品列表 (监听onGoodsUpdate:() 回调更新列表)
@property (nonatomic, strong)NSMutableArray <TFgoods *>* _Nullable goods;
@end

```

进入房间参数的模型

```

@interface TFStart : NSObject
    【字段含义】房间标题
@property (nonatomic, copy, nonnull) NSString* title;

@end

```

直播结束

```

@interface TFStop : NSObject
    【字段含义】房间标题
@property (nonatomic, copy, nonnull) NSString* title;

@end

```

邀请榜单模型

```

@interface TFInvitationListModel : NSObject
    【字段含义】用户唯一 id
@property(nonatomic,strong)NSString * _Nullable xid;
    【字段含义】头像地址
@property(nonatomic,strong)NSString * _Nullable avatar;
    【字段含义】昵称
@property(nonatomic,strong)NSString * _Nullable nickname;
    【字段含义】邀请人数
@property(nonatomic,assign)NSInteger inviteCount;

@end

```

邀请榜单列表

```
@interface TFInvitationListData : NSObject
    //【字段含义】初始化数据错误时的标语
    //@property(nonatomic, strong)NSString * _Nullable msg;
    //【字段含义】初始化数据。为 0 时表示请求成功
    //@property(nonatomic, assign)NSInteger code;
    【字段含义】房间标题
    @property (nonatomic, strong)NSMutableArray <TFInvitationListModel *>* _Nullable list;
@end
```

错误提示

```
@interface TFAsk : NSObject

@end
```

主播的资料

```
@interface TFZhubo : NSObject
    【字段含义】名字
    @property (nonatomic, copy, nonnull) NSString* nickname;
    【字段含义】头像
    @property (nonatomic, copy, nonnull) NSString* avatar;
@end
```

点赞结果

```
@interface TFLikeData : NSObject

    【字段含义】总点赞数
    @property (nonatomic, assign)NSInteger times;

@end
```

关注按钮


```
@interface TFFocus : NSObject
    【字段含义】是否启用
@property (nonatomic, assign)NSInteger enable;
    【字段含义】窗口弹出
@property (nonatomic, assign)BOOL popUp;
    【字段含义】名字
@property (nonatomic, copy, nonnull)NSString * name;
    【字段含义】链接
@property (nonatomic, copy, nonnull)NSString * url;
@end
```

打赏主播开关

```
@interface TFReward : NSObject
    【字段含义】是否启用
@property (nonatomic, assign)NSInteger enable;
    【字段含义】类型
@property (nonatomic, assign)NSInteger type;
    【字段含义】名字 (默认:打赏主播)
@property (nonatomic, copy, nonnull)NSString * name;
@end
```

邀请

```
@interface TFInviteRanking : NSObject
    【字段含义】是否启用
@property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 (默认:邀请)
@property (nonatomic, copy, nonnull)NSString * name;
@end
```

客服私聊开关

```
@interface TFService : NSObject
    【字段含义】是否启用
@property (nonatomic, assign)NSInteger enable;
@end
```

入场提示开关

```
@interface TFEntranceTips : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 ( 默认:入场提示)
    @property (nonatomic, copy, nonnull)NSString * name;
@end
```

叠加人气模式

```
@interface TFAccumulation : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;

@end
```

背景图

```
@interface TFBackground : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】链接
    @property (nonatomic, copy, nonnull)NSString * url;
@end
```

人气

```
@interface TFPopularity : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】人气参数 注: (真实人数*人气参数=人气)
    @property (nonatomic, assign)NSInteger num;
@end
```

邀请榜单开关

```
@interface TFInvitationList : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 ( 默认:邀请榜单)
    @property (nonatomic, copy, nonnull)NSString * name;
@end
```

直播倒计时开关

```
@interface TFCountDown : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 (默认:直播倒计时)
    @property (nonatomic, copy, nonnull)NSString * name;
@end
```

点赞开关

```
@interface TFLike : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
    【字段含义】名字 (默认:直播倒计时)
    @property (nonatomic, copy, nonnull)NSString * name;
@end
```

提问

```
@interface TFQuestion : NSObject
    【字段含义】是否启用
    @property (nonatomic, assign)NSInteger enable;
@end
```

菜单入口

```

@interface TFMenu : NSObject

    【字段含义】是否启用 选项卡
@property (nonatomic, assign)NSInteger enable;

//选项卡数组
@property (nonatomic, strong)NSMutableArray <TFtabsData *>* _Nullable tabsDataList;

    【字段含义】名字 _Nullable _Nonnull (默认:菜单入口)
@property (nonatomic, copy, nonnull)NSString * name;

    【字段含义】提问
@property (nonatomic, strong)TFQuestion* _Nullable question;
    【字段含义】内部 商_城开关
@property (nonatomic, strong)TFStore* _Nullable store;
    【字段含义】提_现
@property (nonatomic, assign)BOOL withdraw;
    【字段含义】切换线路开关
@property (nonatomic, assign)BOOL liveLine;
    【字段含义】个人中心开关
@property (nonatomic, assign)BOOL info;
    【字段含义】红_包开关
@property (nonatomic, assign)BOOL redPackRecord;
    【字段含义】举报开关
@property (nonatomic, assign)BOOL report;
@end

```

观看人数

```

@interface TFNumber : NSObject
    【字段含义】是否启用 (以下模式为可选模式, 不选择则显示正常在线人数)
@property (nonatomic, assign)NSInteger enable;
    【字段含义】人气
@property (nonatomic, strong)TFPopularity * _Nullable popularity;
    【字段含义】是否积累人气
@property (nonatomic, strong)TFAccumulation* _Nullable accumulation;
    【字段含义】名字 (默认:人数)
@property (nonatomic, copy, nonnull)NSString * name;
@end

```

获取生活直播相关配置

```

@interface TFLifePageConfig : NSObject

    【字段含义】背景图片
@property (nonatomic, strong)TFBackground *_Nullable background;
    【字段含义】关注按钮
@property (nonatomic, strong)TFFocus *_Nullable focus;
    【字段含义】打赏主播
@property (nonatomic, strong)TFReward *_Nullable reward;
    【字段含义】邀请榜单开关
@property (nonatomic, strong)TFInvitationList *_Nullable invitationList;
    【字段含义】邀请分享观看开关
@property (nonatomic, strong)TFInviteRanking *_Nullable inviteRanking;
    【字段含义】入场提示
@property (nonatomic, strong)TFEntranceTips *_Nullable entranceTips;
    【字段含义】客服私聊开关
@property (nonatomic, strong)TFService *_Nullable service;
    【字段含义】观看人数
@property (nonatomic, strong)TFNumber* _Nullable number;
    【字段含义】直播倒计时开关
@property (nonatomic, strong)TFCountDown* _Nullable countDown;
    【字段含义】点赞开关
@property (nonatomic, strong)TFLike* _Nullable like;
    【字段含义】标签菜单入口
@property (nonatomic, strong)TFMenu* _Nullable menu;
    【字段含义】外部商_城开关
@property (nonatomic, strong)TFStore* _Nullable store;
    【字段含义】浮动广告开关
@property (nonatomic, strong)TFAd* _Nullable ad;
    【字段含义】浮动广告列表
@property (nonatomic, strong) NSMutableArray <TFAdModel *>* _Nullable adlist;
@end

```

课程数据

```

@interface TFLifeRoomInfo : NSObject

@property (nonatomic, strong)TFLifePageConfig * _Nonnull pageConfig;

    获取当前用户个人数据
@property (nonatomic, strong)TFMember * _Nullable myInfo;

@end

```

数据接口说明

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

进入房间

```
//配置房间参数
TalkfunSDKLiveInitParams *params = [[TalkfunSDKLiveInitParams alloc]init];
params.accessToken = access_token;
params.courseType = TalkfunCourseTypeInteractiveLive;

//TODO:初始化SDK-----
self.talkfunSDK = [TalkfunSDKLive sharedInstance];
//进入房间
[self.talkfunSDK enterRoom:params];
```

进入房间的事件回调

- TalkfunSDKLiveDelegate

```
/**@name 进入房间的事件回调
 * - 如果加入成功,code 会是一个正数 (code > 0),代表加入房间成功)。 error 为null
 * - 如果加入失败,code 会是一个负数 (code < 0),代表进房失败的错误码。 error 有值
 * 调用刷新refreshData: 也会重新回调
 * 进房失败的错误码含义请参见[错误码 TalkfunSDK 文件的TalkfunError 枚举
 */
- (void)onEnterRoom:(NSInteger)code result:(TFError*_Nonnull)error
```

-----下面的代码为过期的方法,不建议使用-----

房间初始化: TALKFUN_EVENT_ROOM_INIT

@参数说明:

参数	类型	描述
roomInfo	NSDictionary	房间数据

roomInfo说明:

mod_theftproof: 水印配置的字典

参数	类型	描述
enable	NSNumber	水印的开关

点播的信息: TALKFUN_EVENT_VOD_INFO

@参数说明:

参数	类型	描述
liveid	NSString	房间ID
title	NSString	房间课题
duration	NSString	视频总时长
views	NSNumber	观看次数
album	NSArray	专辑列表

album参数:

参数	类型	描述
access_token	NSString	进入相应房间
title	NSString	房间课题
duration	NSString	视频总时长
id	NSString	房间ID
img_small	NSString	课程缩略图

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_INFO callback:^(id obj) {
//返回数据

{
    album = (
        {
            "access_token" = zUmNhJjNyQzYhVWNlFzYlZTOyMmY1UGMxI2M2MGN1IGf8xxfdt10iIHd0FmI
scD0zUzMxYjN0Ej0iUWbpR3ZlJnIsIiM1EjM4UjI6ICZphnIscjM6ICZpBnIsIiI6IichRXY2FmIsAj0iIXZk5WZn
JCL3gzNxIjM2YDNxojIlJXawhXZiwSNwgDM1Uj0iQWat92byJCLiYjMxAjMiojIl1WYut2Yp5mIsIiclNXdiojIlx
2byJCLiYjMxAjMiojIk1WdiwyNyojIk12XyVmb0JXyWJCLiIXWpF2Zn1mSxV0UKRzdEB1I6IyclR2bjJye;
            duration = 8697;
            id = 1275085;
            "img_small" = "http://p2.talk-fun.com/live/85/12819/mti6nta5nq/thumb-80x60.jpg";
            title = "\U7ed8\U672c";
        },
        {
            "access_token" = jVzMhFzY3QmNzIGNhJmY4UmN0U2Y1czNkv2NiJWNxQDf8xxfdt10iIHd0FmI
scD0zUzMxYjN0Ej0iUWbpR3ZlJnIsIiM1EjM4UjI6ICZphnIscjM6ICZpBnIsIiI6IichRXY2FmIsAj0iIXZk5WZn
JCL3gzNxIjM2YDNxojIlJXawhXZiwSNwgDM1Uj0iQWat92byJCLiYjMxAjMiojIl1WYut2Yp5mIsIiclNXdiojIlx
2byJCLiYjMxAjMiojIk1WdiwyNyojIk12XyVmb0JXyWJCLicG0pF2Zn1mSpF0UKRzdEB1I6IyclR2bjJye;
            duration = 2027;
            id = 1275105;
            "img_small" = "http://p2.talk-fun.com/live/105/12819/mti6ntewnq/thumb-80x60.jpg";
            title = "\U5b66\U524d\U6559\U80b2\U5b66\U626b\U76f2\U8bfe \U7b2c\U4e09\U8bb2"
        };
    };
    duration = 8697;
    liveid = 1275085;
    title = "\U7ed8\U672c";
    views = 989;
};
}];
```

点播开始: TALKFUN_EVENT_VOD_START

`@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_START callback:^(id obj) {

}];
```

点播开始: TALKFUN_EVENT_VOD_STOP

`@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_STOP callback:^(id obj) {

}];
```

聊天列表: TALKFUN_EVENT_VOD_MESSAGE_APPEND

@参数说明：返回来一个字典key是每个问题的id

参数	类型	描述
qid	NSString	问题ID
liveid	NSString	房间ID
uid	NSString	时间戳
xid	NSString	合作方用户ID
role	NSString	用户角色
nickname	NSString	用户名字
content	NSString	内容
time	NSString	时间
startTime	NSNumber	视频开始了多长时间

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_MESSAGE_APPEND callback:^(id obj) {
//返回数据

(
{
message = "\U9001\U7ed9\U8001\U5e08\Uff1a[S_FLOWER]";
nickname = "\U738b\U6210\U70bc-\U5de5\U5546\U4f01\U4e1a\U7ba1\U7406\U672c\U79d1";
role = user;
starttime = 1824;
timestamp = 1449640957;
xid = 302184;
}
)
}];
```

问题列表: TALKFUN_EVENT_VOD_QUESTION_APPEND

@参数说明：返回来一个字典key是每个问题的id

参数	类型	描述
qid	NSString	问题ID
liveid	NSString	房间ID
uid	NSString	时间戳
xid	NSString	合作方用户ID
role	NSString	用户角色
nickname	NSString	用户名字
content	NSString	内容
time	NSString	时间
startTime	NSNumber	视频开始了多长时间

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_QUESTION_APPEND callback:^(id obj) {
//返回数据

(
{
message = "\U9001\U7ed9\U8001\U5e08\Uff1a[S_FLOWER]";
nickname = "\U738b\U6210\U70bc-\U5de5\U5546\U4f01\U4e1a\U7ba1\U7406\U672c\U79d1";
role = user;
starttime = 1824;
timestamp = 1449640957;
xid = 302184;
}
)
}];
```

章节列表: TALKFUN_EVENT_VOD_CHAPTER_LIST

@参数说明：返回回来的是包含多个字典的数组

参数	类型	描述
course	NSString	课名
page	NSString	页数
starttime	NSString	开始时间
thumb	NSString	课节图片地址

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_CHAPTER_LIST callback:^(id obj) {
//返回数据
(
{
    course = "\U6d3b\U529b\U6fc0\U60c5\U56db\U5c04\U5706\U5708\U5fc3\U5f62\U60c5\U4eba\
U8282PPT\U6a21\U677f - \U526f\U672c";
    page = 1;
    starttime = 0;
    thumb = "http://p2.talk-fun.com/live/4/12541/mti7ndawna/doc/39824bafc5e828a041a1e4f5d
660da03/1_1_s.jpg";
}
)
}];
```

网络速度： TALKFUN_EVENT_NETWORK_SPEED

@参数说明：

参数	类型	描述
speed	NSString	当前速度
networkStatus	TalkfunNetworkStatus	网络状态

```
调用示例： [self.talkfunSDK on:@" network:speed " callback:^(id obj) { //数据 { networkStatus = 0; speed =
"1.1MB/s"; }
```

```
}}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间： 2021-05-25 16:11:05

回放聊天

方法：`getChatWithDuration`

功能描述：按时间点获取对应时间区间的聊天数据列表

@参数说明：

参数	类型	描述
duration	NSInteger	指定时间点(单位秒)

返回参数说明：

参数	类型	描述
uid	NSString	客户用户ID
starttime	NSString	聊天开始的时间戳
gid	NSString	客户定义的用户组ID
avatar	NSString	头像地址(不存在该字段，或者该字段为空，代表没有设置头像，请使用本地默认头像)
role	NSString	用户角色(对应TalkfunSDK.h里面的角色定义)
nickname	NSString	用户昵称
msg	NSString	聊天内容
xid	NSString	该用户在欢拓平台上的唯一ID

调用示例：

```
[talkfunSDK getChatWithDuration:10 callback:^(id obj) {
{
    //返回数据示例
    [{
        uid = "2311926";
        starttime = "536";
        timestamp = "1506425518";
        time = "536";
        gid = "0";
        avatar = "https://lp3-4.talk-fun.com/avatar/229/304/332/16607104.jpg";
        cmd = "1";
        role = "user";
        nickname = "豹哥????-行政管理专科";
        msg = "对";
        xid = "16607104";
    },
    {
        uid = "2308755";
        starttime = "526";
        timestamp = "1506425508";
        time = "526";
        gid = "0";
        avatar = "https://lp3-4.talk-fun.com/avatar/72/537/1153/16611537.jpg";
        cmd = "2";
        role = "user";
        nickname = "王建霞V-行政管理专科";
        msg = "送给老师：[S_FLOWER][S_FLOWER][S_FLOWER]";
        xid = "16611537";
    }
}
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

视频容器

- 只需要在talkfunSDK初始化后传入一个视频容器即可
- 播放时sdk会生成一个视频图添加到该容器
- 也可以在talkfunSDK初始化后,通过- (void)exchangePPTAndCameraContainer 交换ppt容器和摄像头容器

```
- (void)configureCameraContainerView:(UIView *)cameraContainerView;
```

调用示例：

把视频容器给SDK（要显示摄像头区域的必须部分）

```
[self.talkfunSDK configureCameraContainerView:self.cameraView];
```

视频播放时间： TALKFUN_EVENT_VOD_DURATION

@参数说明： 只有一个NSNumber类型的duration

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_VOD_DURATION callback:^(id obj) {  
    //没有返回数据  
}]
```

视频播放时间： TALKFUN_EVENT_PLAY

@参数说明： 没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_PLAY callback:^(id obj) {  
    //没有返回数据  
}]
```

摄像头显示： TALKFUN_EVENT_CAMERA_SHOW

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CAMERA_SHOW callback:^(id obj) {  
    //可以做显示摄像头的操作  
}]
```

摄像头隐藏： TALKFUN_EVENT_CAMERA_HIDE

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_CAMERA_HIDE callback:^(id obj) {  
    //可以做显示摄像头的操作  
}]
```

桌面分享开始: **TALKFUN_EVENT_DESKTOP_START**

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_DESKTOP_START callback:^(id obj) {  
    //没有返回数据  
}];
```

桌面分享结束: **TALKFUN_EVENT_DESKTOP_STOP**

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:TALKFUN_EVENT_DESKTOP_STOP callback:^(id obj) {  
    //没有返回数据  
}];
```

切换视频模式: **TALKFUN_EVENT_MODE_CHANGE**

@参数说明：没有参数

调用示例：

```
[self.talkfunSDK on:@" live:mode:change " callback:^(id obj) {  
    //没有返回数据  
}];
```

...

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

回放问答

方法：**getQuestionWithDuration**

功能描述：按时间点获取对应时间区间的问答数据列表

@参数说明：

参数	类型	描述
duration	NSInteger	指定时间点(单位秒)

返回参数说明：

参数	类型	描述
qid	NSString	问答唯一ID
uid	NSString	客户用户ID
xid	NSString	该用户在欢拓平台上的唯一ID
startTime	NSString	聊天开始的时间戳
gid	NSString	客户定义的用户组ID
avatar	NSString	头像地址(不存在该字段，或者该字段为空，代表没有设置头像，请使用本地默认头像)
role	NSString	用户角色(对应TalkfunSDK.h里面的角色定义)
nickname	NSString	用户昵称
content	NSString	问答内容
answer	NSArray	回复内容，字段名称同上

调用示例：

```
[talkfunSDK getQuestionWithDuration:10 callback:^(id obj) {
{
    //返回数据示例
    [
    {
        answer = (
            {
                content = "可以的";
                qid = "1216382";
                nickname = "邵老师";
                role = "admin";
                uid = "zj_10679347";
                startTime = 4404;
                xid = "16615112";
            },
        );
        content = "我没有高中毕业证，明年我能报考吗";
        qid = "1216372";
        nickname = "孤独的青春-行政管理专科";
        role = "user";
        uid = "2312522";
        startTime = 4303;
        xid = "16615499";
    }
    ]
}];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

点播线路切换

- 调用：
 - - (void)getNetworkList:(void (^)(id result))callback; //获取回放线路
 - - (NSArray *)getNetworkLinesList; //获取数据 (方法已过期, 不建议使用)
 - - (void)setNetworkLine:(NSNumber *)networkLineIndex; //设置线路

@参数说明：数组里面装的是字典，字典里面数据如下：

参数	类型	描述
current	NSNumber	是否当前线路（0不是，1是）
name	NSString	线路名
index	NSNumber	线路指针（调用setNetworkLine传进的参数）

```
//新方法返回数据： isVodLive为YES, 是回放
{
  data = {
    operators = (
      {
        index = 1;
        name = "线路1";
        current = 1;
      },
      {
        index = 2;
        name = "线路2";
        current = 0;
      },
    );
  };
  code = 0;
  isVodLive = 1;
}

{
  current = 1;
  index = 1;
  name = "\U7ebf\U8def1";
},
{
  current = 0;
  index = 2;
  name = "\U7ebf\U8def2";
}
```

```
},  
{  
    current = 0;  
    index = 3;  
    name = "\U7ebf\U8def3";  
}  
)  
}
```

//旧方法返回数据:(

```
{  
    current = 1;  
    index = 1;  
    name = "\U7ebf\U8def1";  
},  
{  
    current = 0;  
    index = 2;  
    name = "\U7ebf\U8def2";  
},  
{  
    current = 0;  
    index = 3;  
    name = "\U7ebf\U8def3";  
}  
)
```

调用示例:

```
NSArray * array = [self.talkfunSDK getNetworkLinesList];  
  
NSNumber * index = array[0][@"index"];  
  
[weakSelf.talkfunSDK setNetworkLine:index];
```

```
//获取线路列表  
NSArray * list = [self.talkfunSDK getNetworkLinesList];  
//拿到相应线路指针  
NSNumber * index = list.firstObject[@"index"];  
//设置线路  
[self.talkfunSDK setNetworkLine:index];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间: 2020-05-13 15:42:51

广告片头设置

```
/**
 * 设置广告视图
 **/
- (void)setADVideoContainer:(UIView*)view;

/**
 * 跳过广告
 **/
- (void)skipAD;
```

广告播放代理

@参数说明： 只有一个NSNumber类型的duration

```
/**
 * 广告播放总时长
 * 是否可跳过广告
 */
- (void)onADPrepare:(TalkfunADConfig*)model ;
/**
 *
 * 广告播放倒计时
 */
- (void)onADCountDownTime:(NSInteger)duration;

/**
 *
 * 广告播放状态改变
 */
- (void)onADVideoStatusChange:(TalkfunMultiMediaStatusChangeListener)loadState;
```

TalkfunADConfig 模型

```
@interface TalkfunADConfig : NSObject
@property (assign, nonatomic) NSInteger duration;//广告总时长
@property (assign, nonatomic) BOOL isSkip;//是否可以跳过广告
@end
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间： 2020-05-13 15:42:51

TALKFUN_EVENT_BROADCAST

```
//TODO: 播放时间同步的广播
[self.talkfunSDK on:TALKFUN_EVENT_BROADCAST callback:^(id obj) {

    NSLog(@"打印广播====>%@", obj);

}];
```

当前课程所有的广播

```
/**参数说明：
 *
 * [
 * {
 *   "status" : "0",
 *   "uid" : "",
 *   "time" : "1865",           时间
 *   "nickname" : "",         名字
 *   "role" : "user",         角色
 *   "msg" : "1",             内容
 *   "cmd" : "3",             cmd=3 公共广播 cmd=4 自定义广播
 *   "a" : 0,
 *   "xid" : "0",
 *   "timestamp" : "1566731562", 时间戳
 *   "gid" : "0"
 * }
 * ]
 */
```

```
- (NSArray*)getBroadcastJSONArray;
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

1.先初始化小班SDK对象

- self.talkfunSDK = [[TalkfunSDKLive alloc] initWithAccessToken:token whiteboard:view configureCameraContainerView:self.cameraView];

2.通过sdk对象操作画板类

- self.talkfunSDK.rtcManager.whiteboard

*

```
#import <Foundation/Foundation.h>
typedef enum: NSUInteger{
    TalkfunSDKWhiteboardTypeStroke, //画笔
    TalkfunSDKWhiteboardTypeCircle, //圆形
    TalkfunSDKWhiteboardTypeRectangle, //矩形
    TalkfunSDKWhiteboardTypeStraightLine, //直线
    TalkfunSDKWhiteboardTypeDottedLine, //虚线
    TalkfunSDKWhiteboardTypeDelete, //框删除
    TalkfunSDKWhiteboardTypeArrow, //箭头
    TalkfunSDKWhiteboardTypeText //文本
}TalkfunSDKWhiteboardType;
//颜色
typedef enum: NSUInteger{
    TalkfunSDKWhiteboardColorTypeBlackColor, //黑
    TalkfunSDKWhiteboardColorTypeRedColor, //红
    TalkfunSDKWhiteboardColorTypeYellowColor, //黄
    TalkfunSDKWhiteboardColorTypeGreenColor, //绿
    TalkfunSDKWhiteboardColorTypeBlueColor, //蓝
    TalkfunSDKWhiteboardColorTypeBrownColor, //棕
    TalkfunSDKWhiteboardColorTypeWhiteColor, //白
    TalkfunSDKWhiteboardColorTypeCyanColor, //青色
    TalkfunSDKWhiteboardColorTypePurpleColor //紫色
}TalkfunSDKWhiteboardColorType;
//画笔大小
typedef enum: NSUInteger{
    TalkfunPlatformDrawLineWidthSmall = 2, //小
    TalkfunPlatformDrawLineWidthMiddle = 4, //中
    TalkfunPlatformDrawLineWidthBig = 6, //大
    TalkfunPlatformDrawLineWidthLarge = 8,
    TalkfunPlatformDrawLineWidthHuge = 10 //特大
}TalkfunPlatformDrawLineWidth;
```

```
@interface TalkfunSDKWhiteboard : NSObject

//涂鸦权限 默认为 NO
@property (nonatomic,assign) BOOL drawPower;

//画板类型 ->
@property (nonatomic, assign) TalkfunSDKWhiteboardType drawType;
//画笔宽度
@property (nonatomic, assign) TalkfunPlatformDrawLineWidth lineWidth;
//画笔颜色(选择的)
@property (nonatomic, assign) TalkfunSDKWhiteboardColorType lineColor;

//rgb的color
- (void)setLineRgbColor:(UIColor*)rgbColor;

@end
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

常见问题

- ipad的横竖屏适配

把Supported interface orientations (iPad)里面的东西都删除

- 进入了我们的SDK之后，其它的网络请求失败

如果进入了我们的SDK后要进行其它的网络请求，要调用-(void)setDomainWhitelist:(NSArray *)list;方法设置url的白名单，否则网络请求可能有误

```
[self.talkfunSDK setDomainWhitelist:@[@"172.16.117.235:7786"]];
```

- 编译通过了但是视频加载的时候出现错误

尝试调换libTalkfunSDK.a文件的位置，将其拖到Linked Frameworks and Libraries的最上端（即变成第一个）

- 通过accessKey鉴权

```
parameters[TalkfunAuthentication] = @(TalkfunAuthenticationByAccessKey);
NSString * accessKey = @"";
//1.实例化SDK (如果用的是accessKey则在access_token参数中传入)
self.talkfunSDK = [[TalkfunSDK alloc] initWithAccessToken:accessKey parameters:parameters];
```

- Demo设计默认竖屏，如需布局上的修改可自行设置，必须的操作是：

- 初始化SDK
- PPT容器
- 摄像头容器

- playbackID

- 作用：区别下载离线文件的唯一标识
- 获取：
 - 可直接用我们的直播ID
 - 可用你们后台生成的ID

- 模拟器支持 用工程里Library/Universal/libTalkfunSDK.a文件替换Library/libTalkfunSDK.a

- 视频容器和ppt容器不能用自动布局，必须设置容器的frame

- 断网后需要调用reload接口刷新

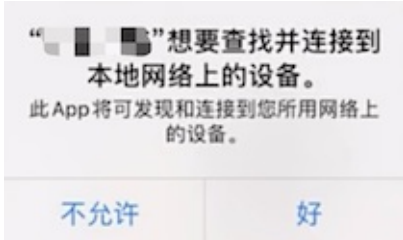
- ffmpeg奔溃问题

◦

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

为什么运行 iOS app 时会看到查找本地网络设备的弹窗提示？

- iOS 系统版本升级至 14.0 版本后，用户首次使用集成了 iOS 语音或视频 SDK 的 app 时会看到查找本地网络设备的弹窗提示。默认弹窗界面如下图所示：



- 解决方案
- 在你的 app 中集成 3.1.2 以下版本 SDK，用户在 iOS 14.0 设备上首次使用 app 时看到的弹窗提示默认为：此 App 将可发现和连接到您所用网络上的设备。建议你结合你的业务需求修改弹窗提示内容，修改步骤如下：

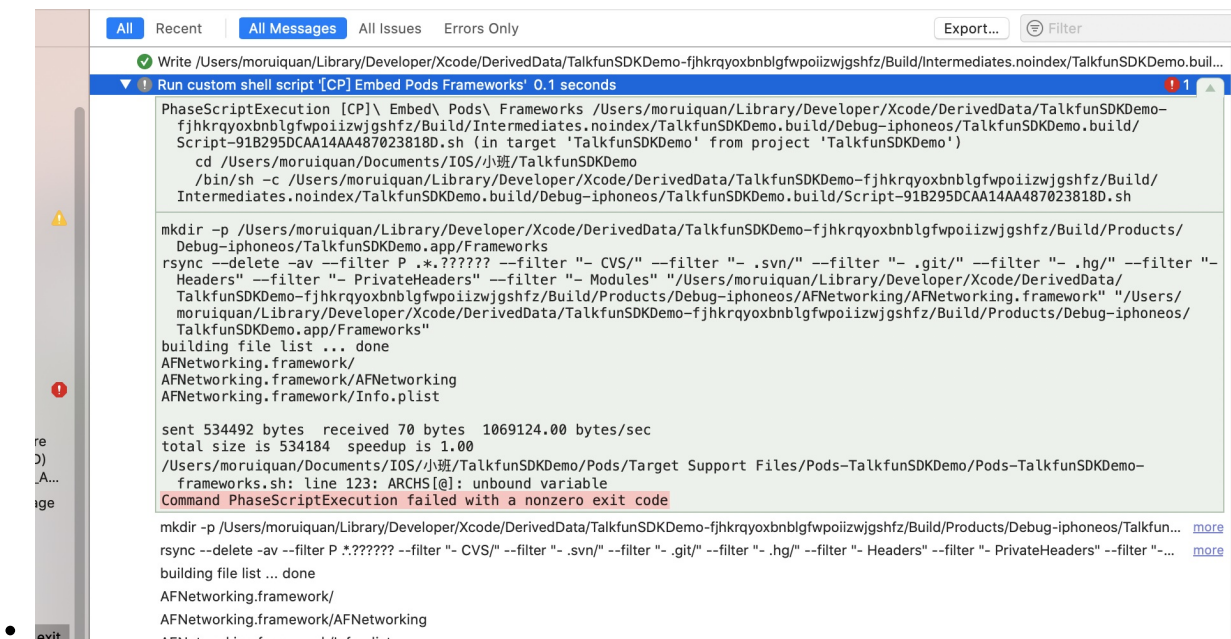
- 1.打开 Xcode 项目，在 info.plist 文件中，点击 + 图标添加 Privacy - Local Network Usage Description。
- 2.结合你的业务需求，在 Privacy - Local Network Usage Description 的 Value 栏中填写获取本地网络设备权限的目的。例如：此 app 不会连接到您所用网络上的设备，只会检测与您本地网关的连通性。
-
- 修改后，用户首次使用 app 时收到的弹窗界面如下图所示：

-
- 如果用户点击好，app 可以通过 reportRtcStats 的 gatewayRtt 参数获取客户端到本地路由器的往返延时。
- 如果用户点击不允许，reportRtcStats 的 gatewayRtt 参数会失效（值恒为 -1），app 无法通过该参数获取客户端到本地路由器的往返延时。
- 用户也可以在 iOS 设备的设置->隐私->本地网络界面修改 app 的权限设置。

- [Privacy - Local Network Usage Description](#)
- [Support local network privacy in your app](#)

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-09-16 15:16:03

32位处理器手机出现错误



- o sdk不再支持armv7s 与 armv7 指令集, Build Settings--选项卡, 在搜索条中搜索--Architectures的支持的arm指令集版本, 删除armv7s armv7
- o 因macOS不再兼容32位应用, xcode也打包不了支持 armv7s 与 armv7 指令集 处理器是32位
- o armv7s : iPhone5 | iPhone5C | iPad4(iPad with Retina Display)
- o armv7 : iPhone3GS | iPhone4 | iPhone4S | iPad | iPad2 | iPad3(The New iPad) | iPad mini | iPod Touch 3G | iPod Touch4

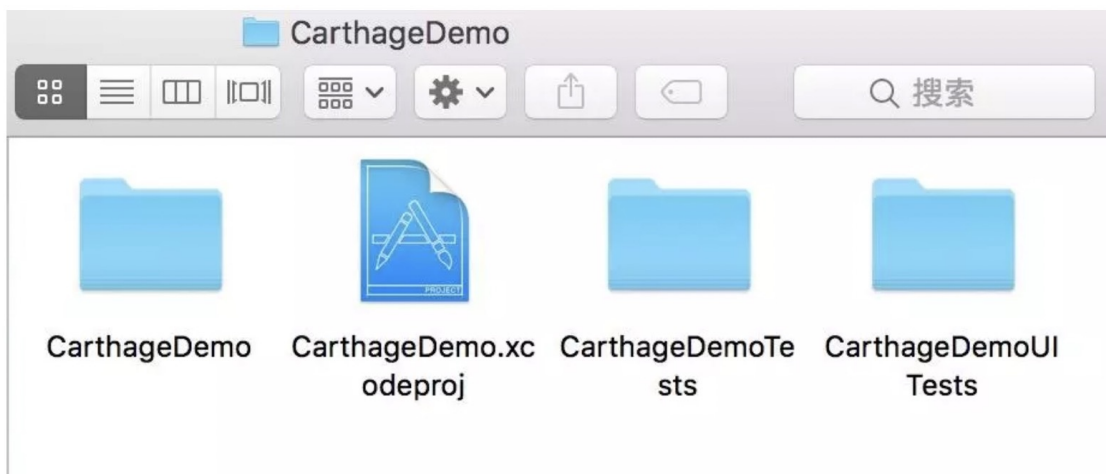
关于项目不能用CocoaPods解决方案

- 主页：[Carthage](#)
- 目标：Carthage旨在使用最简单的方式来管理Cocoa等第三方框架。
- 原理：自动将第三方框架编译为动态库（Dynamic framework）。
- 优点：Carthage为用户管理第三方框架和依赖，但不会自动修改项目文件或构建设置，开发者可以完全控制项目结构和设置，Carthage则更灵活且对项目没有侵入性。。

注意点：Carthage是使用最新版本swift编写,如果Xcode版本过低,要升级为最新版本

其它依赖库oc库可以直接拖到项目中 **Socket.IO-Client-Swift** 是swift库,需使用Carthage导入

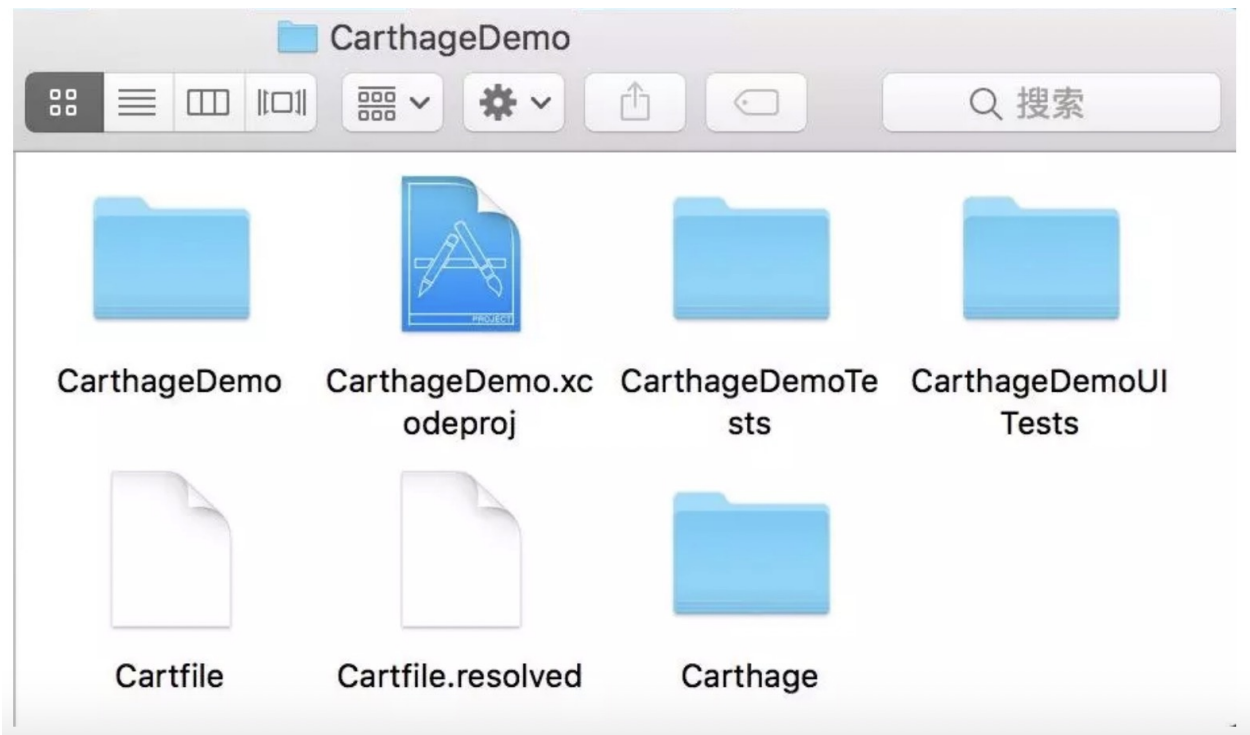
- 1、安装Homebrew，在终端运行：
 - `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
 - 脚本会在执行前暂停，并说明要二次确认回车键继续安装,这条指令运行结束后，Homebrew就安装完成了
- 2、安装Carthage，在终端运行：
 - `brew update`，等待运行结束后，
 - 接着运行：
 - `brew install carthage`，
 - 这样Carthage就安装完成了。
- 3、Carthage的使用
 - 1、新建一个iOS工程



- 2、使用Carthage安装Socket.IO-Client-Swift库
- 3、先进入到项目所在的文件夹 `$ cd ~/Path/CartHageDemo`
- 4、创建一个空的Cartfile文件（两种方式）
 - （1）使用终端创建：`touch Cartfile`

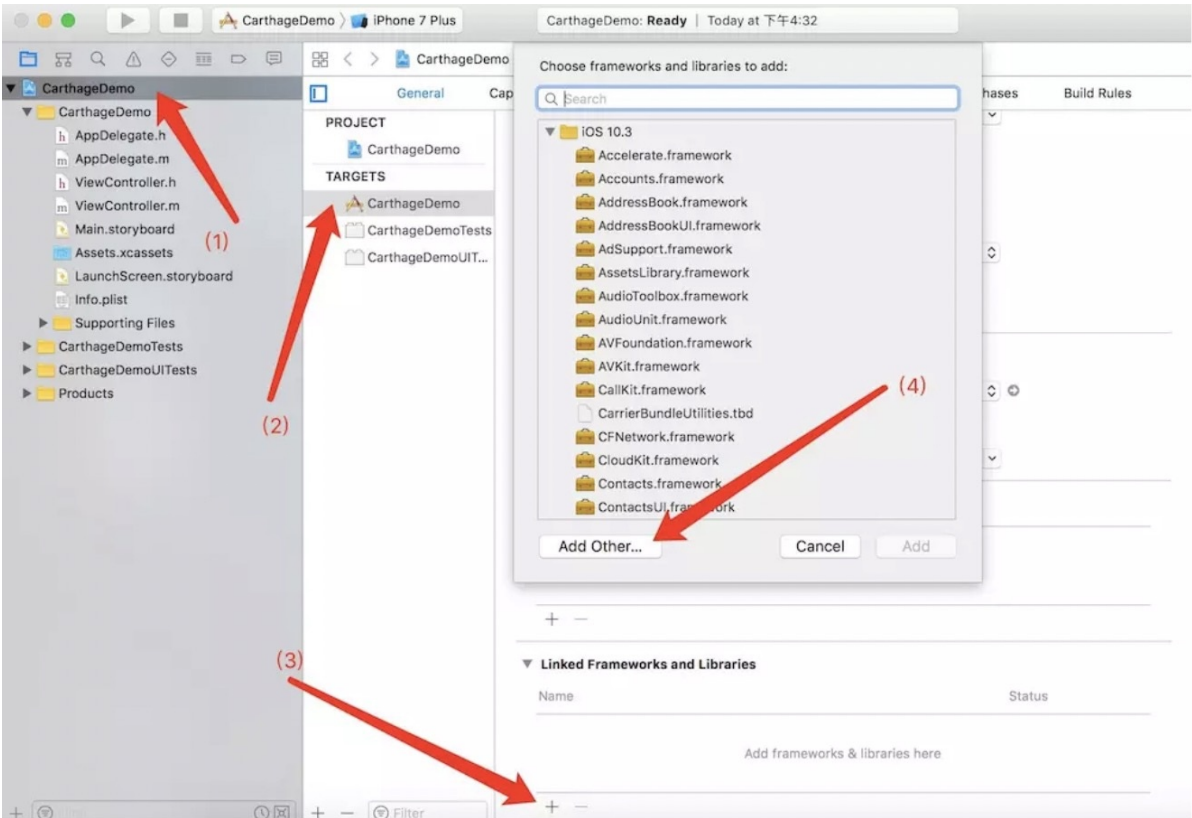
- (2) 使用文本文件创建：创建一个名为Cartfile的文件，并保存到项目目录下
- 5、编辑Cartfile文件
 - github "socketio/socket.io-client-swift" ~> 15.1.0
- 6、保存并关闭Cartfile文件，使用Carthage安装框架 `carthage update`

到此为止，**socketio/socket.io-client-swift** 就已经安装到项目中了

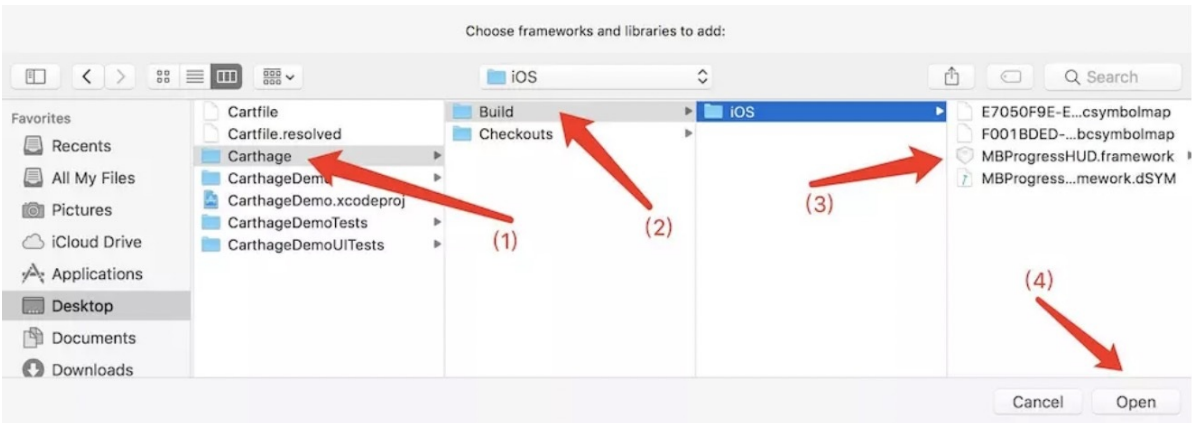


配置项目

- 1、打开项目，点击Target -> General -> Link Library with Libraries选择Carthage/Build目录中导入的framework

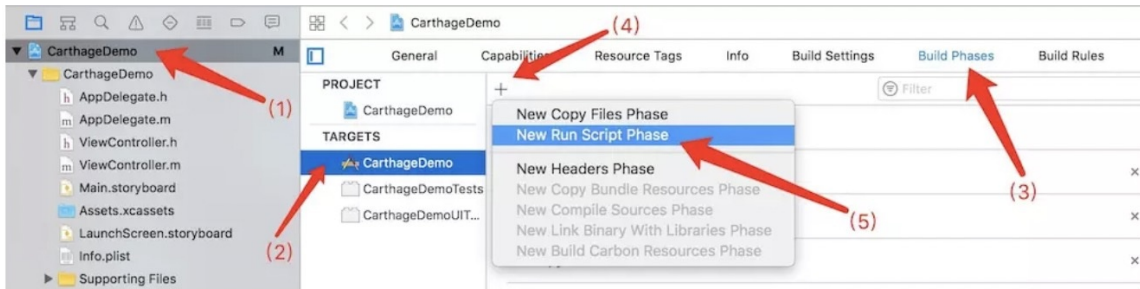


配置项目 (1) .png



配置项目 (2) .png

- 2、添加编译脚本 点击Target -> Build Phases -> “+” -> New Run Script Phase



添加编译脚本 (1) .png

- 3、展开Run Script ,

- 1、在shell下面输入：

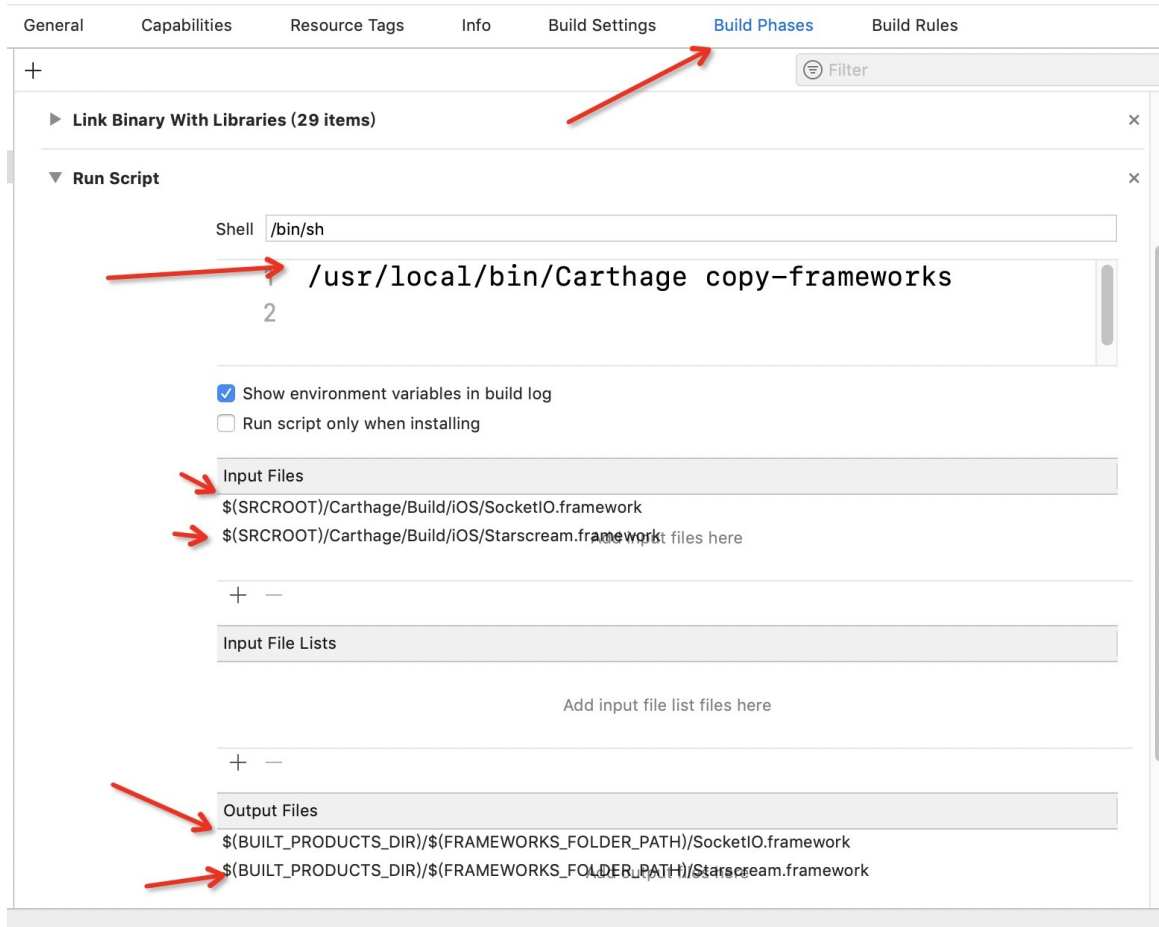
- /usr/local/bin/Carthage copy-frameworks

- 2、在Input Files中加入：

- \$(SRCROOT)/Carthage/Build/iOS/SocketIO.framework
- \$(SRCROOT)/Carthage/Build/iOS/Starscream.framework

- 3、在Output Files中加入

- \$(BUILT_PRODUCTS_DIR)/\$(FRAMEWORKS_FOLDER_PATH)/SocketIO.framework
- \$(BUILT_PRODUCTS_DIR)/\$(FRAMEWORKS_FOLDER_PATH)/Starscream.framework



[Demo传送门] (<http://static-1.talk-fun.com/docs/ios/files/使用Carthage手动导入的SDK.zip>)

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

h5加载的页面内部的视频总是会默认全屏播放的处理

UIWebView直接设置一个属性即可

```
//一个布尔值，允许 H5 中的 Video 标签支持局部视频播放，不全屏视频。  
_webView.allowsInlineMediaPlayback = YES;
```

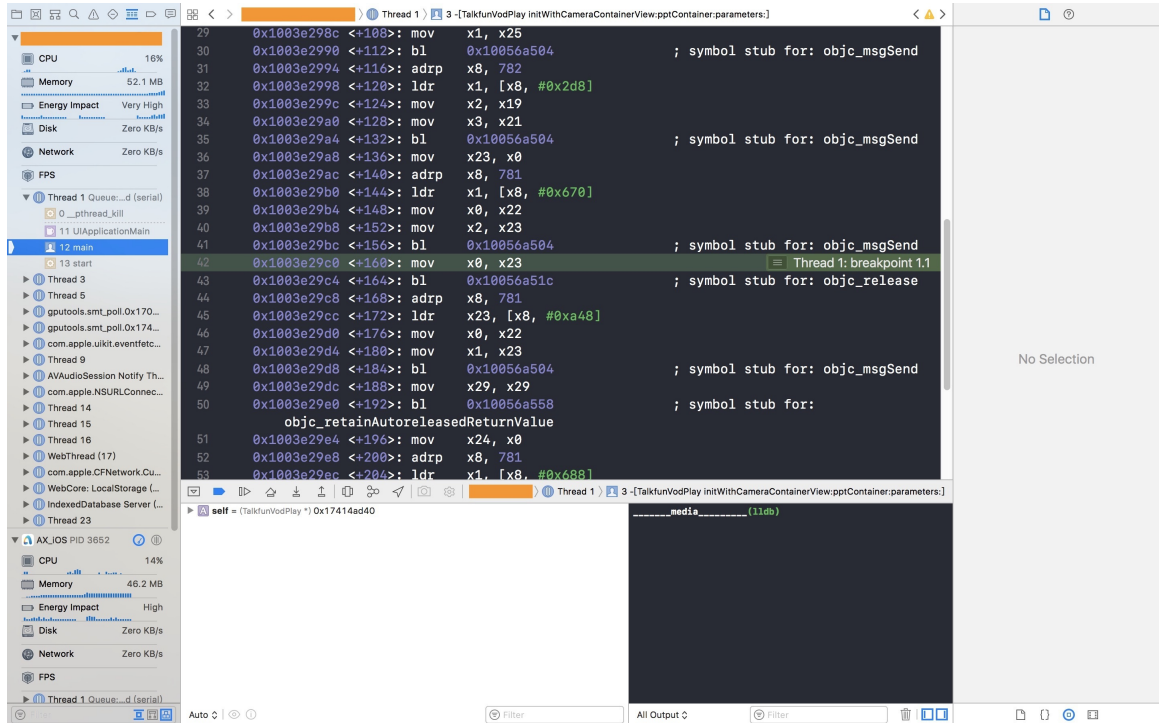
WKWebView配置WKWebViewConfiguration

```
WKWebViewConfiguration *configuration = [[WKWebViewConfiguration alloc] init];  
  
//一个布尔值，允许 H5 中的 Video 标签支持局部视频播放，不全屏视频。  
configuration.allowsInlineMediaPlayback = YES;  
  
WKWebView *webView = [[WKWebView alloc] initWithFrame:CGRectMake(0,20,0,0)configuration:  
configuration];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-12-31 15:44:35

退出直播间重新进入崩溃

- 在退出SDK时要调用一个销毁SDK的方法：- (void) destroy;
 - 没有调用销毁方法会导致SDK销毁不了，出现以下问题：



示例代码：

```
[self.talkfunSDK destroy];
```

Copyright Talkfun all right reserved , powered by Gitbook 修订时间：2020-05-13 15:42:51

横竖屏设置

横竖屏完成由您自己进行设置。

如果需要默认竖屏，将手机方向设置为竖屏，将界面进行竖屏布局即可。

如果需要默认横屏，将手机方向设置为横屏，将界面进行横屏布局即可。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

playbackID的获取及用途

什么是playbackID

在移动原生SDK里面，回放播放功能要求传入一个playbackID参数

playbackID的用途是什么

playbackID主要用于离线播放，离线下载会按照playbackID来创建本地数据的存储文件夹，将该回放的相关数据都下载存储到这个目录下。

当播放回放的时候，SDK首先会检测该目录下是否存在回放的资源文件，如果不存在，将从网络进行加载。

并且，该playbackID会作为离线列表的数据唯一标识，开启、暂停、删除下载都使用该playbackID作为参数。

如何获取playbackID

playbackID参数的主要目的在于，将欢拓的回放数据，映射到本地的文件夹。

*****!!!!!!!重要!!!!!!!*****

那么能够定义一个playbackID与欢拓的回放进行固定的唯一对应关系即可。

这个playbackID，可以使用客户方的课程ID。

另外，调用欢拓API接口创建课程的时候，也会返回一个欢拓的课程ID，

这个ID与客户方的课程ID是一一对应的关系，那么也可以使用欢拓的课程ID。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

Apple Mach-O Linker Error




提供下载的SDK Demo ，默认使用的是 Release-iphone 版本的静态库，支持 armv7、arm64。
如果需要支持模拟器开发，请使用 Universal/libTalkfunSDK.a，支持 i386、armv7、x86_64、arm64。

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51


Failed to create provisioning profile

Provisioning Profile Xcode Managed Profile

Signing Certificate iOS Developer

Status  **Failed to create provisioning profile.**
The app ID "com.talk-fun.SDKDemo" cannot be registered to your development team. Change your bundle identifier to a unique string to try again.

[Try Again](#)

 **No profiles for 'com.talk-fun.SDKDemo' were found**
Xcode couldn't find a provisioning profile matching 'com.talk-fun.SDKDemo'.

SDK Demo默认使用的bundleID为 `com.talk-fun.SDKDemo` 。
使用你们自己的开发者账号对Demo程序进行编译、运行，请修改bundleID为你自己的bundleID。
示例：
`com.xxxxx.SDKDemo`

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

App Transport Security

- 运行中的网络请求出现：

Error Domain=NSURLErrorDomain Code=-1022 "The resource could not be loaded because the App Transport Security policy requires the use of a secure connection."

这样的错误，是由于iOS9改用更安全的https，为了能够在iOS9中正常使用欢拓SDK，请在"Info.plist"中进行如下配置

- v2.3.1版本之前

▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES

- v2.3.1版本之后

- 在下图Exception Domains中添加你要加载url的域名并设置相应的选项即可

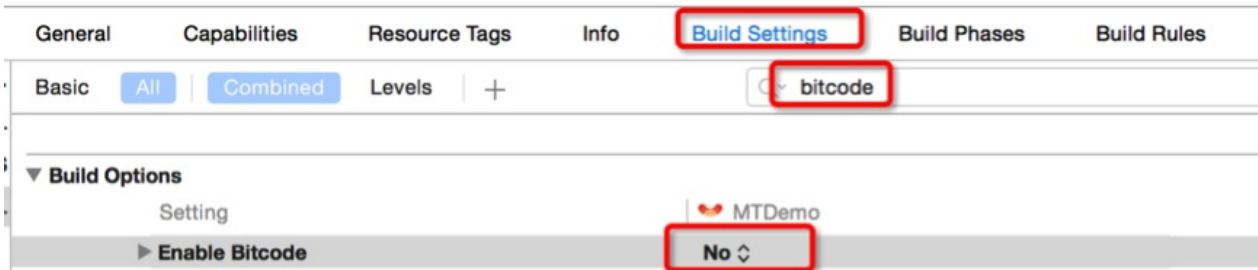
▼ App Transport Security Settings	Dictionary	(2 items)
Allow Arbitrary Loads	Boolean	NO
▼ Exception Domains	Dictionary	(1 item)
▼ talk-fun.com	Dictionary	(3 items)
NSIncludesSubdomains	Boolean	YES
NSEnvironmentAllowsInsecureHTT...	Boolean	YES
NSEnvironmentRequiresForwardSe...	Boolean	YES

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

bitcode

- 编译中出现如下错误 : ld: 'xxx/libTalkfunSDK.a(dict.o)' does not contain bitcode. You must rebuild it with bitcode enabled (Xcode setting ENABLE_BITCODE), obtain an updated library from the vendor, or disable bitcode for this target. for architecture arm64

在Xcode项目里面的 Build Settings -> Build Options -> Enable Bitcode , 设置为 No

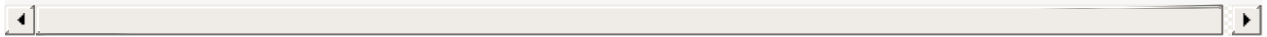


Copyright Talkfun all right reserved , powered by Gitbook修订时间 : 2020-05-13 15:42:51

回放聊天、问答如何自动跟随

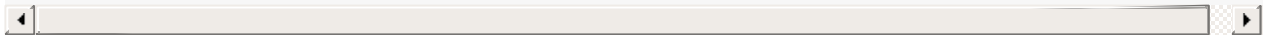
聊天

```
监听 TALKFUN_EVENT_VOD_MESSAGE_REFRESH 获取当前展示的数据区间列表，刷新TableView。  
监听 TALKFUN_EVENT_VOD_MESSAGE_SCROLL 获取当前展示的数据的滚动索引，调整TableView Cell 进行定位。
```



问答

```
监听 TALKFUN_EVENT_VOD_QUESTION_REFRESH 获取当前展示的数据区间列表，刷新TableView。  
监听 TALKFUN_EVENT_VOD_QUESTION_SCROLL 获取当前展示的数据的滚动索引，调整TableView Cell 进行定位。
```



Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

如何在按Home键退到后台后保持继续播放

```
//进入后台是否暂停（默认是暂停）  
[self.talkfunSDK setPauseInBackground:YES];
```

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

如何获取播放缓冲情况

- 直播及回放都存在如下代理方法，在视频缓冲状态改变的时候，SDK会调用该代理方法，具体定义请查看接口文档

/**

- 视频播放状态改变 */

- (void)playerLoadStateDidChange:(TalkfunPlayerLoadState)loadState;

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

关于ReactNative、APICloud或其他非原生语言的接入

- 该SDK使用原生语言编写
- 如果使用的是H5项目，请使用H5相关SDK
- 如果使用非官方原生语言构建APP项目，由于未提供其他非原生SDK，如ReactNative、APICloud等，需要自定义一个中间件，使SDK提供的接口可供项目调用。具体请参考相关语言如何调用原生接口代码

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2020-05-13 15:42:51

SDK错误捕捉 (TalkfunErrorNotification)

- 广播名 : TalkfunErrorNotification //错误信息广播 (类型看错误信息类型)

```
//错误信息类型
typedef enum {

    TalkfunErrorLoadingData = 10000,           //加载数据出错(html、js等)
    TalkfunErrorNetworkDisconnect = 10001,     //网络连接错误
    TalkfunErrorVideoError = 10002,           //视频加载错误
    TalkfunErrorDownloadError = 10003,        //下载错误
    TalkfunErrorLoadingOfflineFileError = 10004 //加载离线文件错误

}TalkfunError;
```

@基本参数说明:

参数	类型	描述
code	TalkfunError	错误码
message	NSString	错误信息

Copyright Talkfun all right reserved , powered by Gitbook修订时间 : 2020-05-13 15:42:51

SDK和文档下载

- SDK版本记录

- Version 4.4.9 (2021-11-16)
 - 新增行为日志上传
 - [SDK下载](#)
- Version 4.4.7 (2021-10-27)
 - 修复已知问题
- Version 4.4.5 (2021-10-15)
 - 修复已知问题
 - 兼容ios15
- Version 4.4.4 (2021-9-29)
 - UI界面依赖的第三方库版本升级
 - 兼容ios15
- Version 4.4.3 (2021-9-15)
 - 修复已知问题,提高稳定
- Version 4.4.0 (2021-8-27)
 - 大班、生活直播支持聊天@回复、删除指定聊天内容
 - 大班回放点播记录播放时间点优化
 - 新表情添加鲜花表情
 - 观看端,大班互动和小班未连麦也能接收涂鸦权限
- Version 4.3.9 (2021-7-26)
 - 支持高度定制化的实时音视频直播方案
 - 修复一些已知问题
- Version 4.3.8 (2021-6-22)
 - 生活直播支持视频插播、桌面分享
 - 支持播放暖场视频
 - 优化虚拟机器人聊天信息显示头像
- Version 4.3.7 (2021-5-28)
 - 优化小班样式
 - 优化生活直播模板
 - 修复一些已知问题
- Version 4.3.4 (2021-5-21)
 - 修复已知问题
- Version 4.3.2 (2021-4-27)
 - 支持主讲模式、画廊模式等视频布局
 - 设备状态优化
- Version 4.3.1 (2021-1-22)

- 支持低延迟大班直播 (需升级依赖库pod 'TalkfunMediaFramework', '2.5.3')
- Version 4.3.0 (2021-1-11)
 - 直播视频播放优化 (需升级依赖库pod 'TalkfunMediaFramework', '2.5.3')
- Version 4.2.8 (2020-12-7)
 - 画板支持显示动态PPT
- Version 4.2.7 (2020-11-24)
 - 桌面模式不断流的优化
- Version 4.2.5 (2020-11-12)
 - 多人提问优化
 - 视频的显示模式优化
- Version 4.2.4 (2020-10-26)
 - 签到优化
 - 离线课程优化
- Version 4.2.3 (2020-10-19)
 - 1v1直播显示优化
 - 登录页优化
- Version 4.2.2 (2020-10-15)
 - 生活直播显示@ 回复
 - 登录页输入密码与名字优化
- Version 4.2.1 (2020-09-27)
 - 性能优化,体验更顺畅
- Version 4.1.0 (2020-09-11)
 - sdk性能优化
 - 数据解码优化
- Version 4.0.1 (2020-09-03)
 - 大班直播聊天显示优化
 - 登录页优化
 - 数据解码优化
- Version 4.0.0 (2020-08-22)
 - 登录页优化
 - 抽奖优化.公告优化.html显示优化
- Version 3.9.9 (2020-08-18)
 - 更改APP名称, 品牌升级, 价值追求以及使命承诺的一次重要升级。
 - 登录页优化
- Version 3.9.6 (2020-07-30)
 - 连麦成功后的涂鸦权限优化
 - 账号被T下线优化
- Version 3.9.5 (2020-07-24)
 - 连接蓝牙的设备连麦优化

- 分组总人数优化
- Version 3.9.4 (2020-07-15)
 - 连麦模式优化
 - 已知问题修复
- Version 3.9.2 (2020-07-02)
 - 支持腾讯RTC
 - 兼容桌面分享模式的时候可以上下讲台,开关讲台
 - 已知问题修复
- Version 3.9.1 (2020-6-28)
 - 1v1.1v16.大班互动. UI界面改版
 - 支持腾讯rtc
 - 回放的播放速率最高支持2.0的倍速
- Version 3.9.0 (2020-6-10)
 - 1v1.1v16.大班互动. UI界面改版
 - 大班直播的抽奖显示改版,显示优化
 - 回放的播放速率最高支持2.0的倍速
- Version 3.8.4 (2020-4-27)
 - 画板PPT显示优化
- Version 3.8.2 (2020-4-22)
 - 大班互动评分功能与课前聊天课后聊天的优化
- Version 3.8.0 (2020-4-15)
 - 大班互动的布局调整
- Version 3.7.9 (2020-4-9)
 - 大班在线人数显示优化
- Version 3.7.8 (2020-3-31)
 - 小班添加统计功能
- Version 3.7.5 (2020-3-17)
 - 优化投票
 - 小班的添加前后摄像头与开关弹幕
- Version 3.7.2 (2012-3-6)
 - 新增互动工具(定时器,抢答,转盘)
- Version 3.7.0 (2012-2-22)
 - 回放聊天显示优化
- Version 3.6.8 (2012-2-11)
 - 画板区域支持显示全屏
 - 视频区域支持显示全屏
- Version 3.6.7 (2012-1-3)
 - 项目去掉UIAlertView
 - 禁言优化

- Version 3.6.6 (2019-12-19)
 - 处理了一些bug
- Version 3.6.4 (2019-12-13)
 - 桌面模式优化
- Version 3.6.1 (2019-12-3)
 - 优化日志
- Version 3.6.0 (2019-11-14)
 - 直播间的设置线路优化
- Version 3.5.0 (2019-11-6)
 - 登录界面优化
 - 启动图显示优化
- Version 3.4.6 (2019-10-28)
 - 直播的音视频与涂鸦同步优化
 - 禁用黑暗模式
- Version 3.4.5 (2019-10-21)
 - 下载回放的资源过期的优化
- Version 3.4.4 (2019-9-26)
 - 1、适配ios 13
- Version 3.4.3 (2019-9-11)
 - 1、视频播放优化
 - 2、修复一些Bug
- Version 3.4.2 (2019-8-28)
 - 1、支持回放播放片头视频
 - 2、支持回放广播
- Version 3.4.1 (2019-8-15)
 - 1、支持被邀请上讲台
 - 2、支持播放音视频素材
- Version 3.4.0 (2019-7-16)
 - 1、文字交互配置
 - 2、动态评分配置
 - 3、新增白板页区域回调接口
 - 4、UI层优化
- Version 3.3.4 (2019-6-6)
 - 新增1对1直播模式
 - 修复已知bug
- Version 3.3.3 (2019-4-14)
 - 优化画板
 - 优化下载

- 修复已知bug
- Version 3.3.1 (2019-2-14)
 - 课程回放支持弹幕
 - 课程回放记录播放时间
 - 修复已知bug
- Version 3.3.0 (2019-2-14)
 - 班课互动和视频连麦模式功能优化
 - 修复已知bug
- Version 3.2.3 (2019-1-3)
 - 优化画板涂鸦的优化
 - 修复已知bug
- Version 3.2.0 Version 3.2.0 (2019-1-3)
 - 优化画板涂鸦的优化
 - 统计优化
 - 直播观看人数增加机器人数量
 - 增加防盗录水印显示
 - 修复已知bug
- Version 3.1.0 (2018-9-11)
 - 优化画板涂鸦的优化
 - 新增都是老师的评分功能
 - 修复已知bug
 - 新增首次进入直播间能看到历史聊天功能
- Version 3.0.7 (2018-6-14)
 - 优化下载进度
 - 优化画板 (修复word文档滚动涂鸦位置偏移)
 - 修复已知bug
 - socket连接加sessionid和xid参数
- Version 3.0.5 (2018-4-12)
 - 日志模块优化
 - seek进度失败
 - 画板优化
 - 缓存课程模块优化
- Version 3.0.4 (2018-1-9)
 - 支持后台下载回放
 - 异常崩溃保存下载进度，重新启动后可以断点续传
 - 删除正在下载的回放Bug修复
 - 新增图片铺满功能
 - SDK内部日志优化
 - SDwebImage更新到4.0
 - 直播的网络选择,修复了网络线路数大于2显示不正确的问题

- Version 3.0.3 (2017-11-9)
 - 新增接收签到，提交签到
 - 新增显示在线访客人数
 - 新增获取所有投票与新的投票接口
 - 新增获取所有广播与新的广播接口
- Version 3.0.2 (2017-10-12)
 - 点播章节UITableView滚动时监测视频播放进度的监听事件TALKFUN_EVENT_VOD_DURATION不回调的问题
 - 修复点播画板切换ppt时闪一下的问题
 - 支持直播、点播自动切换线路
 - 优化缓存点播发生错误时的通知
 - 修复已知bug
- Version 3.0.0 (2017-09-18)
 - sdk重构，性能提升
 - 画板渲染性能提升，提升显示清晰度
 - 直播、点播头文件分离
 - 修复已知bug
- Version 2.3.4 (2017-05-11)
 - 聊天提问增加头像显示
 - 投票增加答案显示
 - 处理表情与文字高度不对的问题
 - 点播多段视频平滑切换
 - 新增initWithAccessKey接口
 - 新增appendDownloadWithAccessKey接口
- Version 2.3.3
 - 皮肤界面修改
 - 表情更换
 - 增加图片投票功能
 - 增加鉴权方式
 - 优化SDK加载
- Version 2.3.2
 - 修复已知问题
 - 优化离线播放
- Version 2.3.1
 - 离线播放优化
 - SDK错误信息捕捉
 - 点播线路切换
 - 点播倍速播放
 - 支持https

- 新增独立视频播放接口
- Version 2.3.0
 - 优化代码结构，接入更加清晰
- Version 2.2.8
 - 1.增加进入后台继续播放设置
 - 2.增加SDK log打印控制
 - 3.增加url白名单设置
 - 4.修复已知bug
- Version 2.2.7

Copyright Talkfun all right reserved , powered by Gitbook修订时间：2021-11-17 10:40:21