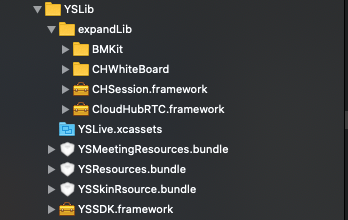
YSSDK For iOS 集成手册

V3.3.3.0

1. SDK工程结构



YSSDK.framwork: UI支持SDK

YSLive.xcassets: 图片资源

ExpendLib: 需要引用的库包括:

1. BMKit.framework: 基础库
2. CloudHubRTC.framework: 音视频SDK
3. CHSession.framework: 业务SDK
4. CHWhiteBoard.framework: 白板SDK
5. YSWhiteBoardResources.bundle: 白板资源文件
6. YSResources.bundle: 文本资源文件
7. 将以上文件添加到工程中
8. 配置工程文件
9. 工程配置修改(Build Settings)
   1. Build Optins->Enable Bitcode->修改为NO

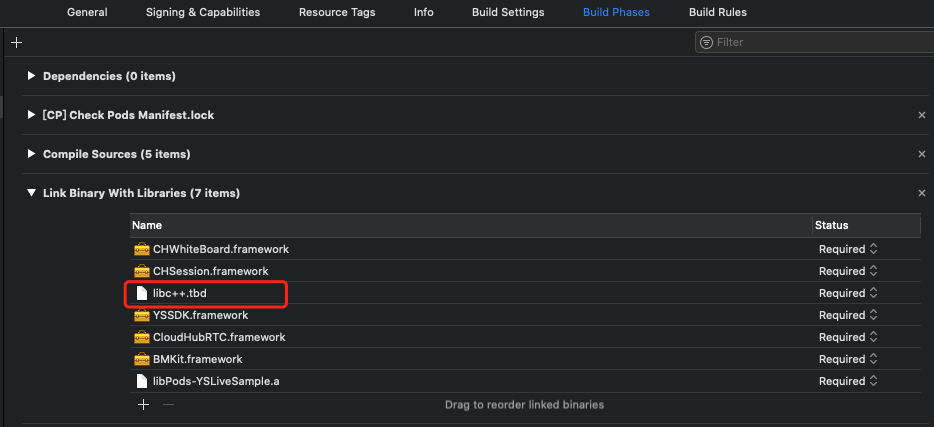


* 1. Linkink->Other Linker Flage添加

-all\_load 和 -ObjC

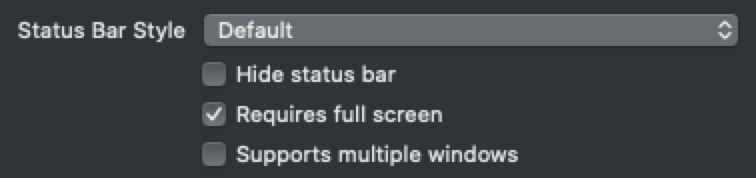


1. Build Phases->Link Binary With Libaries 添加必要依赖库: libc++.tbd



1. 工程支持旋转方向设置，请尽量参照sample来设置旋转

方法一：General->Device Orientation设置添加横屏方向Landscaoe Right



方法二：请参照sample在AppDelegate中添加转屏代码，建议使用此方法

- (UIInterfaceOrientationMask)application:(UIApplication \*)application supportedInterfaceOrientationsForWindow:(UIWindow \*)window

{

if (self.allowRotation)

{

return UIInterfaceOrientationMaskLandscapeRight;

}

else

{

return UIInterfaceOrientationMaskPortrait;

}}

1. 将CloudHubRTC.framework 拷贝到工程中

方法1：直接将CloudHubRTC.framework的 Embed选项设置为Embed & Sign



方法2：保持CloudHubRTC.framework 的Embed选项为Do Not Embed默认值

然后在Build Phases添加New Copy Files Phase项



将CloudHubRTC.framework加入



1. 语言设置 info.plist -> Localization native development region->China

请根据自己需求设置语言，目前SDK支持中文简体，中文繁体和英文

1. 设置权限

info.plist -> Privacy - Camera Usage Description -> 房间中需要进行视频通话以及拍摄您是否允许打开相机

info.plist -> Privacy - Microphone Usage Description -> 房间中需要发送语音消息及发言您是否允许打开麦克风

info.plist -> Privacy - Photo Library Usage Description -> 房间中需要选择本地图片您是否允许访问相册

info.plist -> Privacy - Photo Library Additions Usage Description -> 房间中需要上传图片您是否允许添加图片

info.plist ->App Transport Security Settings->Allow Arbitrary Loads->YES

或将以下代码加入info.plist

<key>NSAppTransportSecurity</key>

<dict>

<key>NSAllowsArbitraryLoads</key>

<true/>

</dict>

<key>NSCameraUsageDescription</key>

<string>房间中需要进行视频通话以及拍摄您是否允许打开相机</string>

<key>NSLocationWhenInUseUsageDescription</key>

<string>房间中需要通过您的地理位置信息获取您周边的位置相关数据您是否允许开启位置</string>

<key>NSMicrophoneUsageDescription</key>

<string>房间中需要发送语音消息及发言您是否允许打开麦克风</string>

<key>NSPhotoLibraryAddUsageDescription</key>

<string>房间中需要上传图片您是否允许添加图片</string>

<key>NSPhotoLibraryUsageDescription</key>

<string>房间中需要选择本地图片您是否允许访问相册</string>

<key>UIBackgroundModes</key>

<array>

<string>audio</string>

</array>

1. 代码调用
2. 导入头文件并初始化

#import <YSSDK/YSSDKManager.h>

@interface YSLoginVC ()

<

YSSDKDelegate

>

@property (nonatomic, weak) YSSDKManager \*ysSDKManager;

@end

- (void)viewDidLoad

{

[super viewDidLoad];

self.ysSDKManager = [YSSDKManager sharedInstance];

[self.ysSDKManager registerManagerDelegate:self];

}

1. 检查房间类型

\_\_weak \_\_typeof(self) weakSelf = self;

[self.ysSDKManager checkRoomTypeBeforeJoinRoomWithRoomId:roomId success:^(YSSDKUseTheType roomType, BOOL needpassword) {

// roomType: 房间类型 3：小班课 4：直播 6：会议

// needpassword: 参会人员(学生)是否需要密码

if (self->userRole == YSSDKSUserType\_Student)

{

// 学生登入

// 注意： 直播只支持学生身份登入房间

[weakSelf.ysSDKManager joinRoomWithRoomId:roomId nickName:nickName roomPassword:nil userId:nil userParams:nil];

}

else

{

// 老师(会议主持)登入

// 注意： 小班课和会议支持老师和学生身份登入房间

[weakSelf.ysSDKManager joinRoomWithRoomId:roomId nickName:nickName roomPassword:nil userRole:self->userRole userId:nil userParams:nil];

}

} failure:^(NSInteger code, NSString \* \_Nonnull errorStr) {

NSLog(@"code:%@, message: %@", @(code), errorStr);

[self.progressHUD hideAnimated:YES];

}];

1. 进入房间

// 学生登入

// 注意： 直播只支持学生身份登入房间

[weakSelf.ysSDKManager joinRoomWithRoomId:roomId nickName:nickName roomPassword:nil userId:nil userParams:nil];

// 老师(会议主持)登入

// 注意： 小班课和会议支持老师和学生身份登入房间

[weakSelf.ysSDKManager joinRoomWithRoomId:roomId nickName:nickName roomPassword:nil userRole:self->userRole userId:nil userParams:nil];

1. 状态回调

#pragma mark -

#pragma mark YSSDKDelegate

/\*\*

成功进入房间

@param ts 服务器当前时间戳，以秒为单位，如1572001230

@param roomType 房间类型

@param userType 登入用户身份

\*/

- (void)onRoomJoined:(NSTimeInterval)ts roomType:(YSSDKUseTheType)roomType userType:(YSSDKUserRoleType)userType;

/\*\*

失去连接

\*/

- (void)onRoomConnectionLost

{ NSLog(@"onRoomConnectionLost");

}

/\*\*

已经离开房间

\*/

- (void)onRoomLeft

{ NSLog(@"onRoomLeft");

}

/\*\*

自己被踢出房间

@param reason 被踢原因

\*/

- (void)onRoomKickedOut:(NSDictionary \*)reason

{ NSLog(@"onRoomKickedOut");

}

/\*\*

发生密码错误 回调

需要重新输入密码

@param errorCode errorCode

\*/

- (void)onRoomNeedEnterPassWord:(YSSDKErrorCode)errorCode

{ NSLog(@"onRoomNeedEnterPassWord");

}

/\*\*

发生其他错误 回调

需要重新登陆

@param errorCode errorCode

\*/

- (void)onRoomReportFail:(YSSDKErrorCode)errorCode descript:(NSString \*)descript

{ NSLog(@"onRoomReportFail");

}

/\*\*

已经进入直播房间

\*/

- (void)onEnterLiveRoom;

/\*\*

已经进入小班课(会议)房间

\*/

- (void)onEnterClassRoom;

回调- (void)onRoomReportFail:descript:中的

错误编码YSSDKErrorCode参照 YSSDKDefine.h 中的定义

1. 房间需要重新输入密码时的操作

- (void)roomManagerNeedEnterPassWord:(YSSDKErrorCode)errorCode

{

NSLog(@"roomManagerNeedEnterPassWord");

// 重新设置密码password后调用

[self.ysSDKManager joinRoomWithRoomId:roomId nickName:nickName roomPassword:password userId:nil userParams:nil];

}

1. 补充一些屏幕旋转的说明

由于本SDK在运行时会将屏幕旋转，在退出房间后返还前置页面窗口时需要还原屏幕方向，需要注意以下几点：

1. 根据自己的UI架构选择适配方法

// 根控制器是导航控制器，那么在这个导航控制器中实现下面三个方法

-(BOOL)shouldAutorotate {

return [[self.viewControllers lastObject] shouldAutorotate];

}

-(NSUInteger)supportedInterfaceOrientations {

return [[self.viewControllers lastObject] supportedInterfaceOrientations];

}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {

return [[self.viewControllers lastObject] preferredInterfaceOrientationForPresentation];

}

// 根控制器是tabBar控制器，那么在这个tabBar控制器中实现下面三个方法

-(BOOL)shouldAutorotate {

return [self.selectedViewController shouldAutorotate];

}

-(NSUInteger)supportedInterfaceOrientations {

return [self.selectedViewController supportedInterfaceOrientations];

}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {

return [self.selectedViewController preferredInterfaceOrientationForPresentation];

}

以上都是为了将控制权转交给UIViewController，需要在你的ViewController实现屏幕方向设置

例如：竖屏

// 竖屏方向

- (BOOL)shouldAutorotate {

return NO;

}

- (UIInterfaceOrientationMask)supportedInterfaceOrientations {

return UIInterfaceOrientationMaskPortrait;

}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {

return UIInterfaceOrientationPortrait;

}

1. AppDelegate里面的屏幕方向最好也控制一下

- (UIInterfaceOrientationMask)application:(UIApplication \*)application supportedInterfaceOrientationsForWindow:(UIWindow \*)window

{

if (self.allowRotation) {

// 这是我们SDK内部的旋转方向

return UIInterfaceOrientationMaskLandscapeRight;

} else {

return UIInterfaceOrientationMaskPortrait;

}

}