YICameraSDK 文档说明-iOS

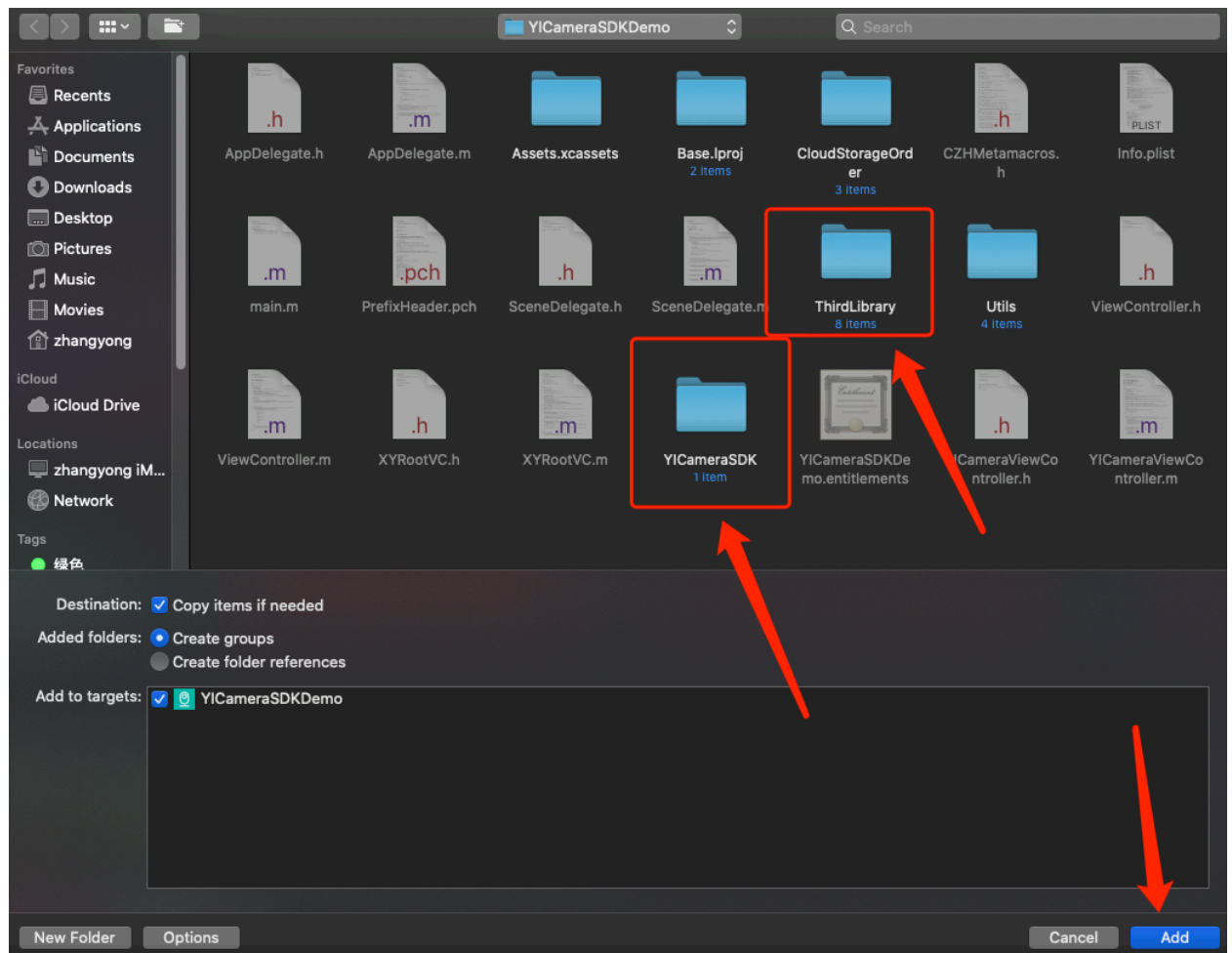# 1. YICameraSDK 使用说明:

1). Xcode 最低支持 iOS 10.0

2). iPhone 最低支持 iPhone 5
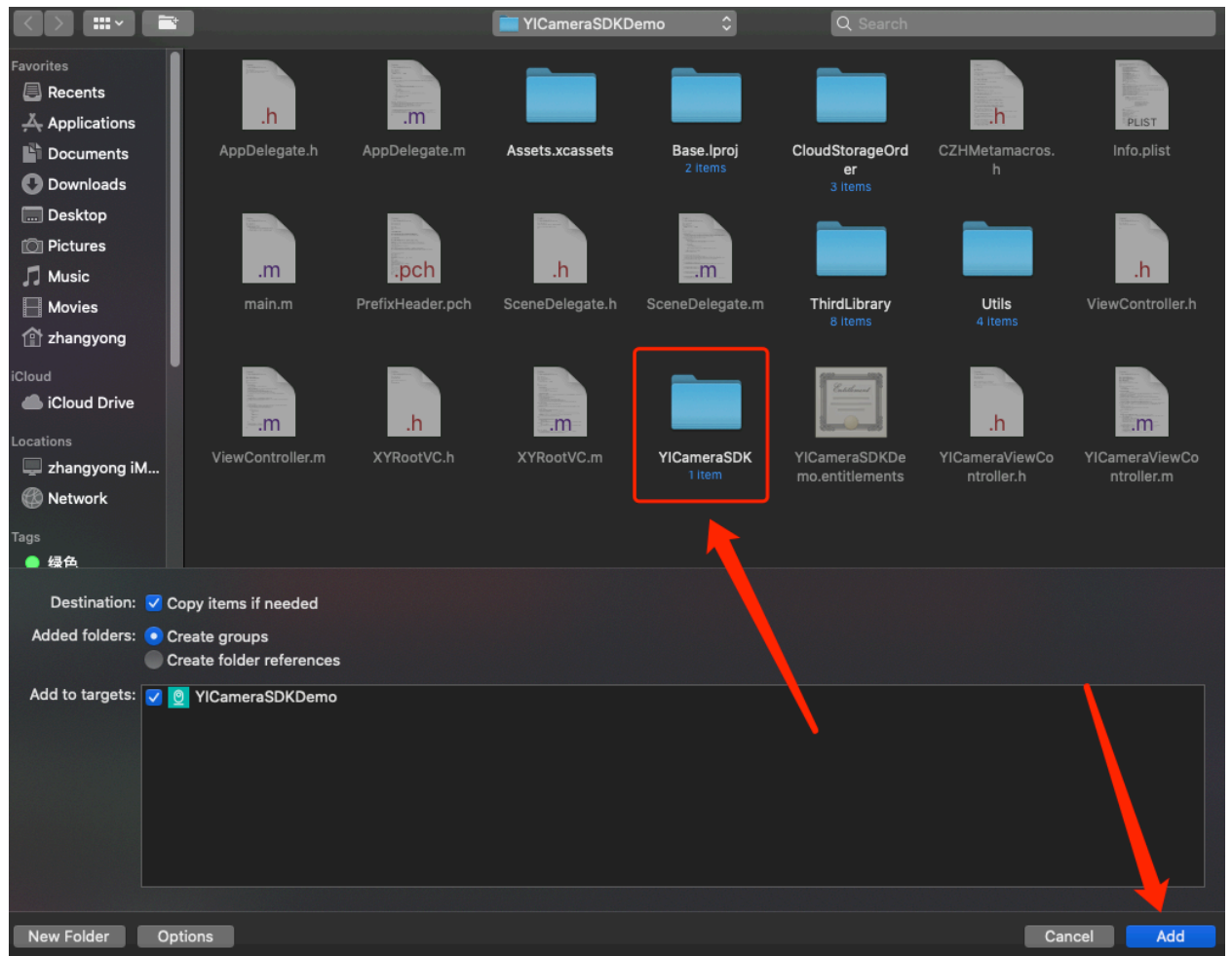
2. 集成文档

1). 导入 YICameraSDK

1.1, 如果你的工程使用 Carthage 管理: 导入 YICameraSDK & ThirdLibrary 文件到工程

1.2，如果你的工程使用 CocoaPods 管理：导入 YICameraSDK 文件到工程

2).三方依赖文件

2.1，如果你的工程使用 Carthage 管理： Cartfile

github "hackiftekhar/IQKeyboardManager"

github "CoderMJLee/MJRefresh"

github "SnapKit/Masonry" == 1.1.0

github "CoderMJLee/MJExtension" == 3.0.17

github "ReactiveCocoa/ReactiveObjC" == 3.1.1

github "AFNetworking/AFNetworking" == 4.0.1

github "SDWebImage/SDWebImage"

github "shu223/PulsingHalo"

github "ccgus/fmdb"

2.2,　如果你的工程使用　CocoaPods 管理：　Podfile

pod 'IQKeyboardManager'

pod 'iCarousel'

pod 'MJRefresh'

pod 'Masonry', '1.1.0'

pod 'MJExtension', '3.0.17'

pod 'dsBridge', '3.0.6'

pod 'ReactiveObjC', '~> 3.1.1'

pod 'AFNetworking', '~> 4.0.1'

pod 'SDWebImage'

pod 'Reachability'

pod 'NVHTarGzip'

pod 'PulsingHalo'

pod 'LRRouter'

pod 'FMDB'

pod 'LKDBHelper'

pod 'SwipeView'

pod 'AMTumblrHud'

3).导入系统依赖库

CoreLocation.framework

CFNetwork.framework

SystemConfiguration.framework

MobileCoreServices.framework

QuartzCore.framework

CoreGraphics.framework

AVFoundation.framework
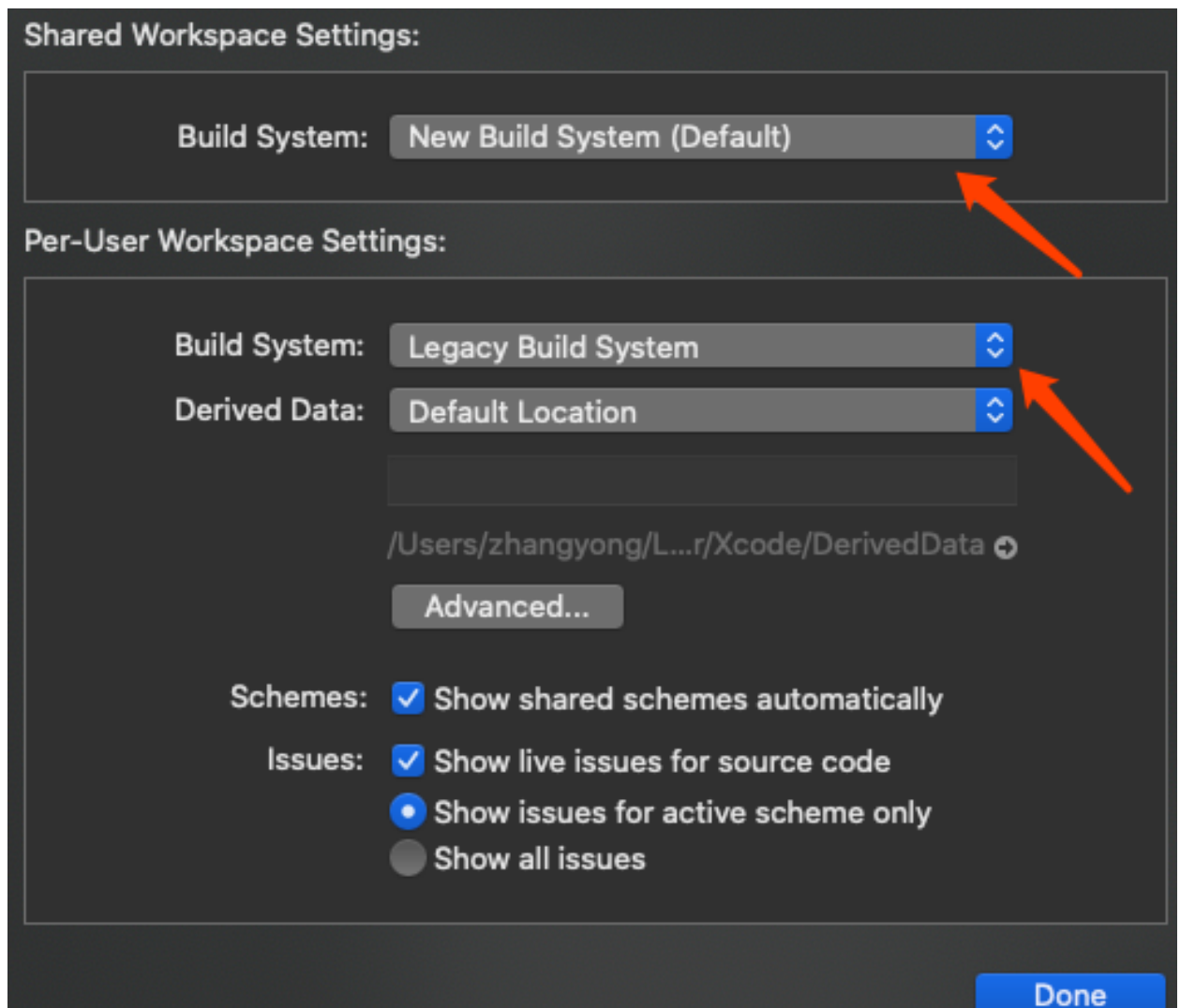
CoreMotion.framework

Photos.framework

ImageIO.framework

libsqlite3.tbd

libiconv.tbd

libc++.tbd

libbz2.tbd

4). Xcode → File →Workspace Settings

5). 添加 wifi 支持

TARGETS → signing & Capabilities → Access WiFi Information

Notice:Login to your Apple Developer website to set it up

Apple Developer -> Identifiers -> Access WiFi Information

6). Xcode info.plist

App Transport Security Settings → Allow Arbitrary Loads → YES

Privacy - Camera Usage Description

Privacy - Location Usage Description

Privacy - Location When In Use Usage Description

Privacy - Microphone Usage Description

Privacy - Photo Library Additions Usage Description

Privacy - Photo Library Usage Description

3. API 说明

1). YICameraSDK 初始化平台

/// Registration platform

/// @param appId appId

+ (void)registPlatform:(NSString *)appId;

2). 设置用户信息 openId & userToken

/// Binding user information

/// @param openId openId

/// @param userToken userToken

/// @param completeBlock success errorCode

+ (void)setUserInfo:(NSString *)openId token:(NSString *)userToken
complete:(nullable void (^)(BOOL success, NSString * _Nullable
errorCode))completeBlock;

3). 打开绑定页面

/// Open the bound device page

/// @param currentController Current controller

+ (void)openDeviceBindViewComtroller:(UIViewController *)currentController;

4). 打开设备列表页面

/// Open the device list page

/// @param currentController Current controller

+ (void)openDeviceListViewController:(UIViewController *)currentController;

5).打开报警消息列表页面

/// Open alarm message list page

/// @param currentController Current controller

+ (void)openAlsertsViewController:(UIViewController *)currentController;

6).根据设备 uid 打开直播页面

/// Open play page based on device uid

/// @param currentController Current controller

/// @param uid device uid

/// @param completeBlock result

+ (void)openCameraPlayViewController:(UIViewController *)currentController uid:(NSString *)uid complete:(nullable void (^)(BOOL isOpenSuccess, NSString *errorMessage))completeBlock;

7). 根据设备 uid 打开设置页面

/// Open camera set page based on device uid

/// @param currentController Current controller

/// @param uid device uid

/// @param completeBlock result

+ (void)openCameraSetViewController:(UIViewController *)currentController uid:(NSString *)uid complete:(nullable void (^)(BOOL isOpenSuccess, NSString *errorMessage))completeBlock;

8). 根据设备 uid 打开报警列表页面

/// Open alarm details page based on device uid

/// @param currentController Current controller

/// @param uid device uid

/// @param alramTime Timestamp

/// @param completeBlock result

+ (void)openDeviceAlarmViewController:(UIViewController *)currentController uid:(NSString *)uid alramTime:(double)alramTime complete:(nullable void (^)(BOOL isOpenSuccess, NSString *errorMessage))completeBlock;

9). 获取设备列表 json 数据

/// Get device list data

/// @param successBlcok Successfully returned device json data

/// @param failureBlcok Failed return

+ (void)getDeviceList:(nullable void (^) (id _Nullable responseObject))successBlock failure:(nullable void (^)(NSError * _Nullable error))failureBlock;

10).获取 4g 设备 iccid

/// get 4g sim info

/// @param uid device uid

/// @param block iccid Block

+ (void)get4gInfoWithUid:(NSString *_Nonnull)uid iccidBlock:(nullable void(^)(NSString * _Nullable iccid, BOOL success))block;

11).获取绑定设备信息

/// Bind device result

/// @param block uid name

+ (void)onBindDeviceResult:(void(^)(BOOL success, NSString *uid, NSString *name))block;