



上海澎博财经资讯有限公司
Pobo Financial Information Co., Ltd.

博易 APP SDK
iOS 接入指南

Confidential

更改历史/ Revision History

说明：

- 1) 版本序号的编制方法为, 按顺序依次增加, 初始版本为 1.0。当版本排序到 1.9 时, 再次更改后, 版本序号更换到 2.0, 后续排列序号依此类推。
 - 2) 日期的命名按照年-月-日的顺序, 具体格式见上表的示例。

目录

1	文档概述	5
1.1	目的 (Target)	5
1.2	适用范围	5
2	产品概述	6
2.1	产品描述	6
2.1.1	主要功能 (可参考“博易 APP 使用说明书”)	6
2.1.2	SDK 模块关系图	7
2.2	技术支持/版权申明	8
2.2.1	技术支持	8
2.2.2	版权申明	8
3	SDK 接入指南	9
3.1	SDK 目录说明	9
3.2	iOS SDK 集成	10
4	SDK 使用基本流程	14
4.1	方案一 (框架模式) 使用流程	14
4.1.1	启动 SDK	14
4.1.2	设置根控制器及相关属性	14
4.1.3	设置代理	15
4.1.4	自定义资源 copy (可选)	15
4.1.5	消息通知	16
4.1.6	App 跳转打开	17
4.1.7	屏幕旋转	17
4.2	方案二 (单页面模式) 使用流程	17
4.2.1	在 APP 启动时初始化 SDK	18
4.2.2	实现状态回调代理协议接口	19
4.2.3	SDK 启动完成, 创建主导航页面: 行情、交易等	21
4.2.4	注册和接收消息通知	22
4.2.5	SDK 主题(黑白皮肤)切换回调	24
4.2.6	自动横竖屏控制	24
4.2.7	分享相关处理	24
4.2.8	云认证接口	24
4.3	澎博 SDK 接口使用	25
5	澎博 SDK 中涉及第三方 SDK 集成说明	26
5.1	穿透式监管 SDK	26
5.2	分享 SDK (微信、QQ)	26
5.3	百度统计 SDK	28
5.4	云屋 SDK	28
5.5	极光一键登陆	28
5.6	通付盾 OCR SDK	28
5.7	证通 OCR SDK	28
5.8	天威云 OCR SDK	29
6	附录 (Appendix)	29

6.1	SDK 接口定义	29
6.2	SDK 功能相关配置定义	29
6.3	博易 APP 使用说明	29
6.4	博易 APP 测试报告	29
7	备注	30
7.1	删除 XCode11 新建项目中的 SceneDelegate.....	30

Confidential

1 文档概述

1.1 目的 (Target)

本文档的目的是详细介绍博易 APP SDK iOS 端接入操作说明。任何超出本规范对产品功能和规格的需求都须经过上海澎博财经资讯有限公司确认后方可变更。

1.2 适用范围

本文档适用于博易 APP 系统产品整个生命周期。

2 产品概述

2.1 产品描述

博易 APP 是一款集行情、交易、资讯为一体的移动终端软件，界面简洁、功能齐全。博易 APP 提供客户端 SDK，用于实现行情、交易等相关功能。

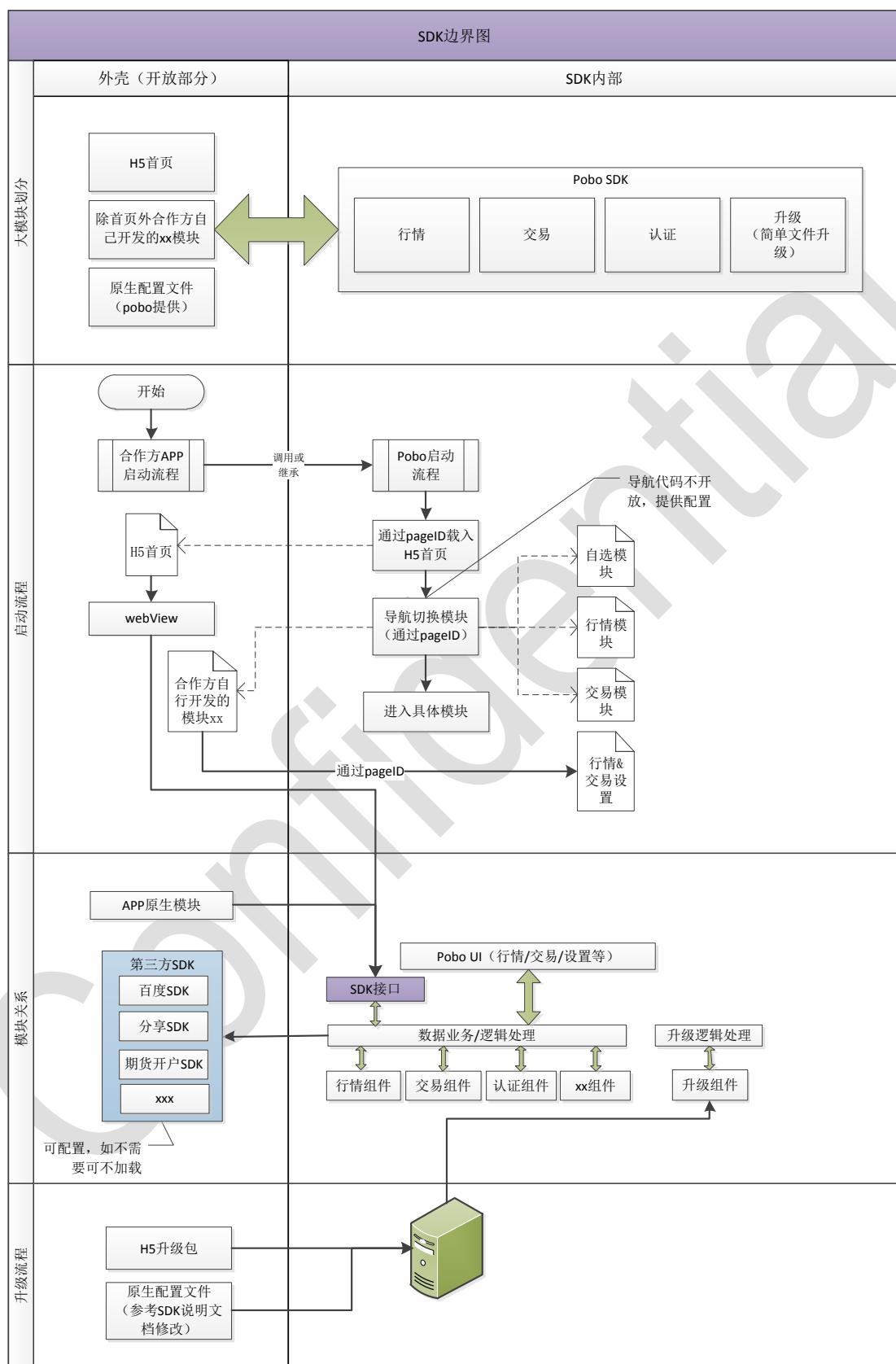
以下内容如无特别指明，均指客户端 SDK；

博易 APP 客户端 SDK 支持的平台有：Android、iOS

2.1.1 主要功能（可参考“博易 APP 使用说明书”）

- 1) 行情报价列表展示（支持多交易所行情）：期货合约、期货期权合约、股票期权合约等等。
- 2) 行情合约详情展示：分时图、K 线图等等。
- 3) 自选：添加、编辑、删除，支持多图同列等。
- 4) 交易：包括三键下单、传统下单、条件单、画线下单、止损止盈、止损开仓、条件单附带止损止盈、银期转账、行权、结算单查询、修改密码等功能。
- 5) 交易----快捷下单。
- 6) 系统设置。
- 7) 行情相关设置。
- 8) 交易相关设置。

2.1.2 SDK 模块关系图



补充说明：

1. 澄博 SDK 所依赖的第三方 SDK（百度&分享&OCR&开户等）放在外壳里，合作方可根据需要确定是否打包入版本，如不需要可删除，对 APP 其他功能无影响。另外，合作方如需要也可以自己调用第三方 SDK。
2. 澄博 SDK 开放的接口不是针对 H5 的，H5 必须通过 WebView（合作方自行开发）才能调用。
3. 澄博 SDK 支持的升级功能目前只支持简单文件升级，不支持 APP 版本升级，建议合作方自建升级机制。

2.2 技术支持/版权申明

2.2.1 技术支持

在您使用 SDK 的过程中，遇到任何困难，请与我们联系，我们将热忱为您提供帮助。
您可以通过如下方式与我们取得联系：

1. 公司官网：<http://www.pobo.net.cn/>
2. 电子邮件/QQ 号码：
3. 服务热线：

2.2.2 版权申明

博易 APP SDK 是上海澎博财经资讯有限公司开发，拥有自主知识产权的产品。本产品禁止反编译分析，在未经上海澎博财经资讯有限公司授权下，不得以任何形式或借助任何手段复制本产品的任何部分。

随同 SDK 包一同发布的 Demo 程序源代码版权归上海澎博财经资讯有限公司。

3 SDK 接入指南

3.1 SDK 目录说明

--PbMobileSDK ----- 博易 SDK 必须库、资源
--PbMobileSDK.framework ----- SDK 代码库
--PbMobileSDKBundle.bundle ----- SDK 相关配置图片等（可修改）
--PbMobilePublicBundle.bundle ----- SDK 代码库依赖的图片、nib 等（基本不可修改，如需修改请联系我们开发人员）
 --rings ----- 声音文件相关（针对系统设置回报类提示声音，此文件不可放到 bundle 里面）

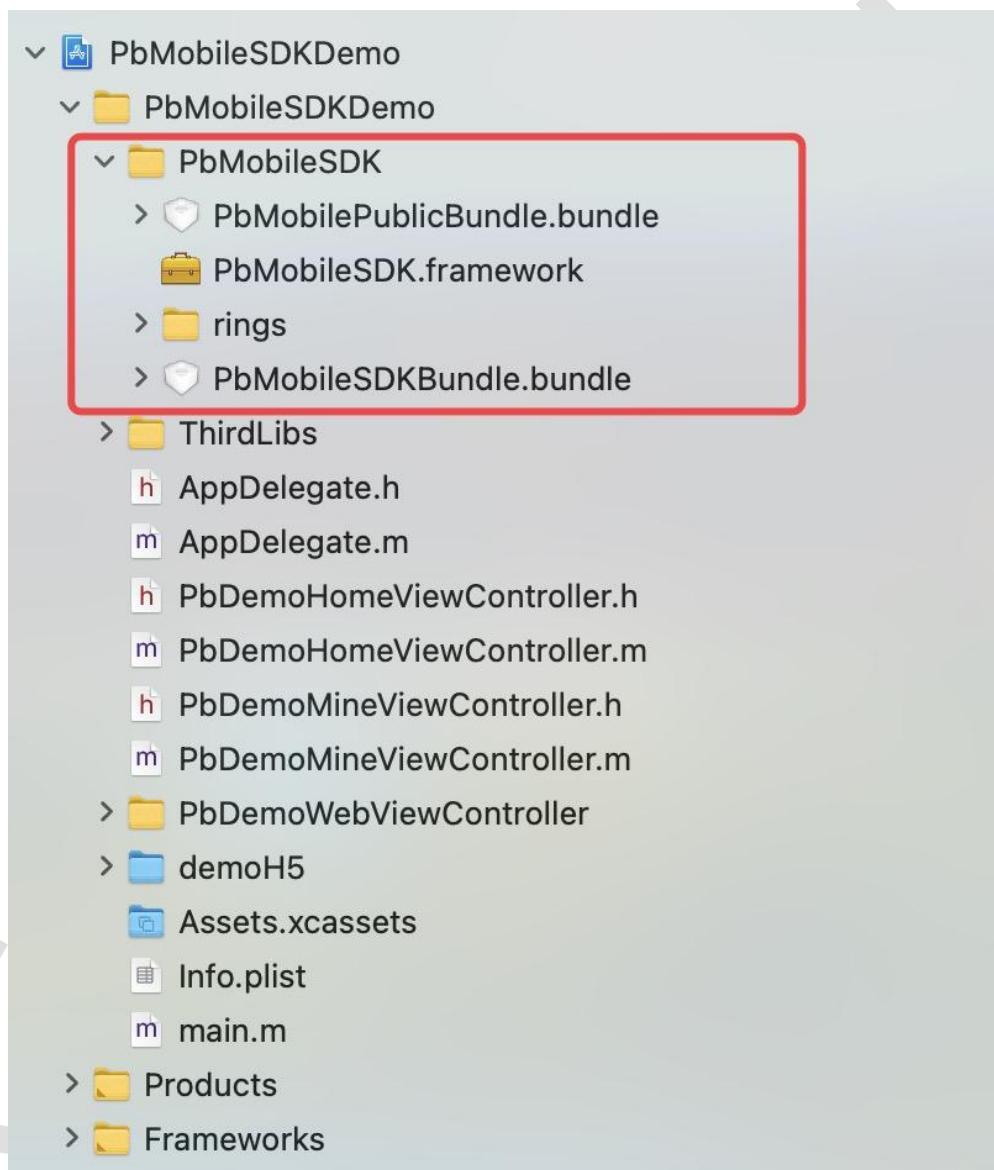
--ThirdLibs ----- 第三方 SDK 库
 -- ChuanTouJianGuan
 -- FeiMa ----- 飞马穿透式
 -- CTP ----- CTP 穿透式库
 -- JinShiDa ----- 金仕达穿透式库
 -- AXSystemInfoLib ----- 顶点穿透式库
 -- HsInfoCollect ----- 恒生穿透式库
 -- ShareSDK ----- 分享 SDK 库
 -- BaiduMobStat ----- 百度统计 SDK 库
 -- VideoCall ----- 云屋 SDK 库
 -- TelecomLogin ----- 极光一键登陆 SDK 库
 -- OCRSDK ----- 通付盾 OCR SDK 库
 -- CWResource.bundle ----- 通付盾 OCR bundle
 -- ZTOCRSDK ----- 证通 OCR SDK 库
 -- TWOCRSDK ----- 天威云 OCR SDK 库

--PbMobileSDKDemo ----- demo 工程

3.2 iOS SDK 集成

(由于 XCode 版本差异，有些配置或代码可能不完全一样，遇到问题请联系[我们](#))

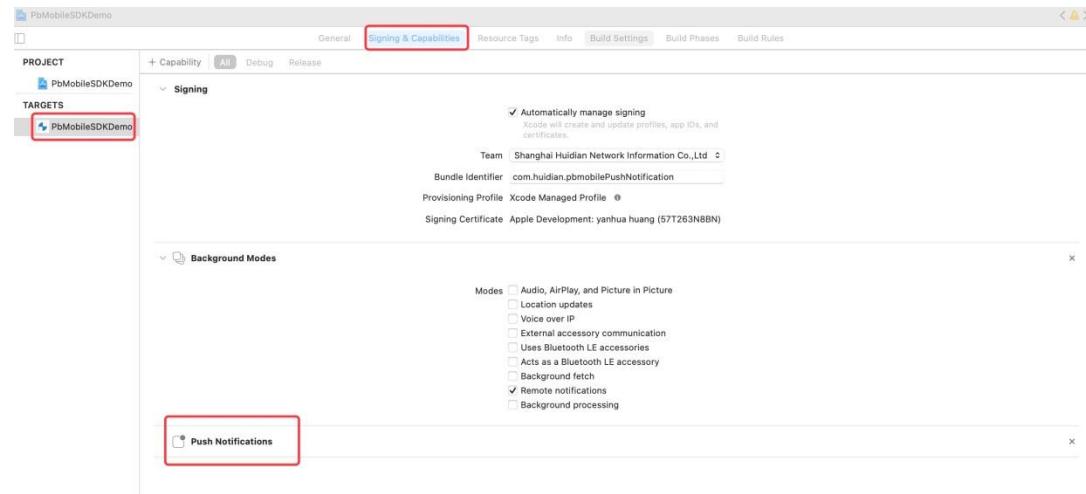
1. 下载 SDK。
2. 创建一个新的 Xcode 工程。
3. 将下载的 SDK 对应的 PbMobile 加入到工程里面，PbMobile 文件夹里面包含四个文件



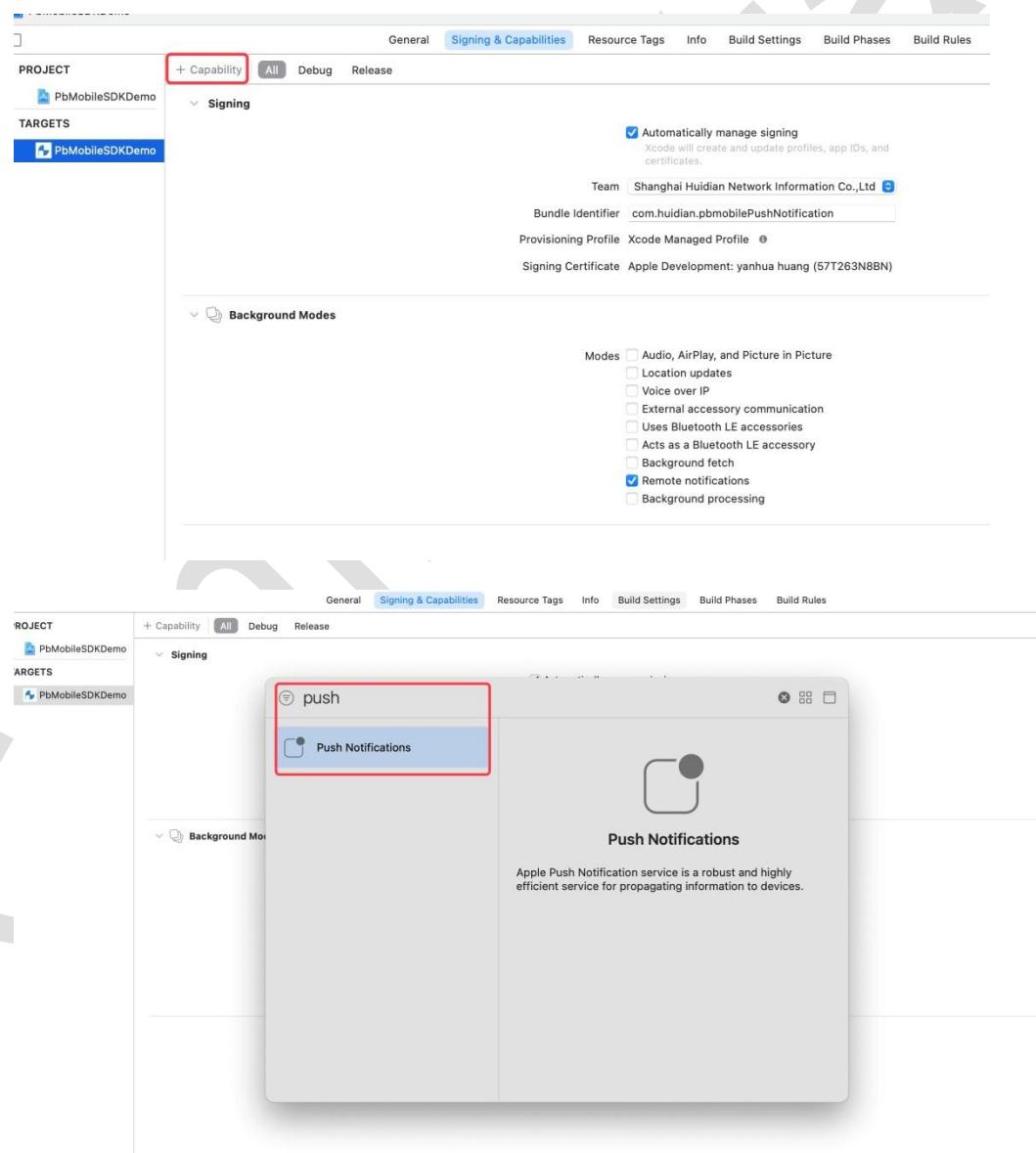
4. 添加完成后，修改相关系统配置、添加系统依赖库

- 1) 添加支持系统离线推送

Signing & Capabilities 选项下确保 Push Notifications 存在

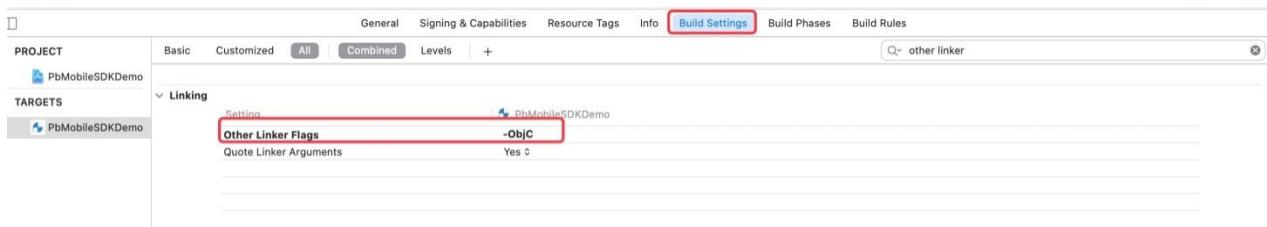


如果没有点击左上角+Capability 搜索添加



2) 添加系统编辑参数

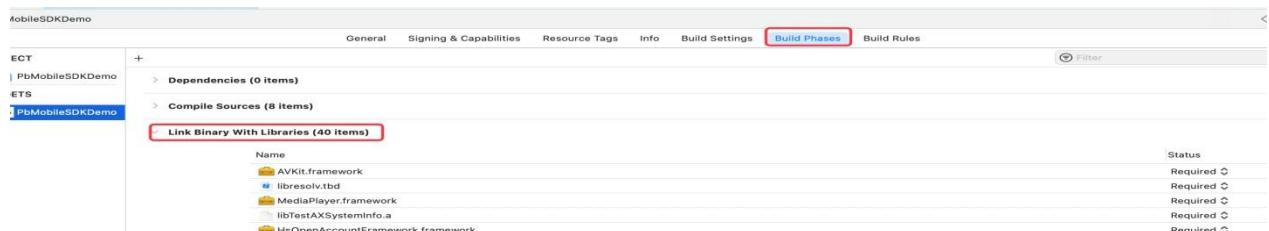
Build Settings 项目下 Other Linker Flags 里面添加 -ObjC (注意大小写)



3) 添加系统依赖库

Build Phases 项目下 Link Binary With Libraries 下面加入依赖的系统库，需要加入的系统库有：

- libresolv.tbd
- libc++.tbd
- libiconv.2.tbd
- libsqlite3.tbd
- libz.1.tbd
- AVKit.Framework
- VideoToolbox.Framework
- WebKit.Framework
- LocalAuthentication.Framework
- MediaPlayer.Framework
- AddressBookUI.Framework
- SystemConfiguration.Framework
- SafariServices.Framework
- ImageIO.Framework
- CoreImage.Framework
- Foundation.Framework
- AVFoundation.Framework
- UIKit.Framework
- QuartzCore.Framework
- AudioToolbox.Framework
- CoreVideo.Framework
- CoreMedia.Framework
- CoreGraphics.Framework
- CoreAudio.Framework
- CoreLocation.Framework
- CoreTelephony.Framework
- Security.Framework
- UserNotifications.Framework
- MessageUI.Framework
- AppTrackingTransparency.framework
- AdSupport.framework



4) 系统权限添加

Info.plist 文件里面需要添加相关系统权限，具体 value 值 demo 里面仅供参考，可自行斟酌使用

相机 —— NSCameraUsageDescription

指纹/面容 —— NSFaciIDUsageDescription

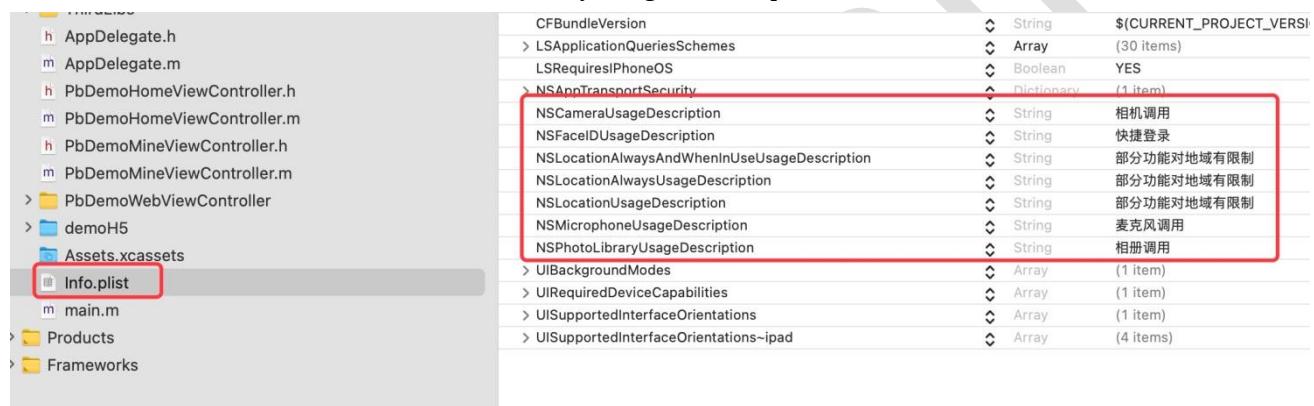
定位 —— NSLocationAlwaysAndWhenInUseUsageDescription

NSLocationAlwaysUsageDescription

NSLocationUsageDescription

麦克风 —— NSMicrophoneUsageDescription

相册 —— NSPhotoLibraryUsageDescription



4 SDK 使用基本流程

4.1 方案一（框架模式）使用流程

整体流程截图

```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    [self.window makeKeyAndVisible];
    //1.启动SDK
    PbMobileManager *mobileManager = [PbMobileManager shareManager];
    [mobileManager startWithOptions:launchOptions];

    //2.设置根控制器
    UIViewController *rootVC = [mobileManager rootViewController];
    //创建UINavigationController并指定根控制器
    UINavigationController *navCtr = [[UINavigationController alloc] initWithRootViewController:rootVC];
    //隐藏navigationBar
    navCtr.navigationBar.hidden = YES;
    //设置该属性让系统时间栏文字变成白色
    navCtr.navigationBar.barStyle = UIBarStyleBlack;
    //设置window的跟控制器为 UINavigationController
    self.window.rootViewController = navCtr;

    //3.设置代理（实现跳转认证）
    [mobileManager setDelegate:self];

    //4.资源拷贝（拷贝自定义H5资源文件到document目录下，可选功能，按照此方式copy过去的，可以用我们的升级平台进行升级）
    [self copyResources];

    //5.注册消息通知
    [self registerRemoteNotification];
    return YES;
}

```

4.1.1 启动 SDK

在 APP 入口 AppDelegate 中

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)la
unchOptions
```

初始化澎博 SDK 并启动

```
PbMobileManager *mobileManager = [PbMobileManager shareManager];
[mobileManager startWithOptions:launchOptions];
```

4.1.2 设置根控制器及相关属性

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)la
unchOptions
```

方法中调用完 SDK 初始化启动后

1) 设置根控制器

```
UIViewController *rootVC = [mobileManager rootViewController];
```

2) 创建 UINavigationController 并指定根控制器

```

UINavigationController *navCtr = [[UINavigationController alloc] initWithRootViewController:rootVC];
3) 隐藏 navigationBar (SDK 里面自定义实现了 titleview)
    navCtr.navigationBar.hidden = YES;
4) 设置该属性让系统时间栏文字变成白色
    navCtr.navigationBar.barStyle = UIBarStyleBlack;
5) 设置 window 的根控制器为 UINavigationController
    self.window.rootViewController = navCtr;

```

4.1.3 设置代理

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions 方法中，设置 PbMobileManager 的代理，并根据需要实现相关代理
[mobileManager setDelegate:self];

代理方法：

1) 打开认证页面（需要认证的必须实现）
- (void)loadAppCertifyView:(UIViewController *)sender;

可在此方法里自定义实现认证相关页面，如使用我们的认证，实现如下：

```
[[PbMobileManager shareManager] openViewController:sender urlString:@"pobo:pageId=800001"];
```

```

#pragma mark - PbMobileManagerDelegate
// 打开认证页面
- (void)loadAppCertifyView:(UIViewController *)sender
{
    [[PbMobileManager shareManager] openViewController:sender urlString:@"pobo:pageId=800001"];
}

```

2) tabbarcontroller 回调（详细使用可查看《澎博手机 SDK 接口文档(iOS)》）
- (void)tabbarDidSelectViewController:(UIViewController *)viewController;
3) H5 回调传递消息（详细使用可查看《澎博手机 SDK 接口文档(iOS)》）
- (void)webViewCallback:(UIViewController *)viewController eventObj:(NSString *)eventObj;

4.1.4 自定义资源 copy (可选)

SDK 支持外部资源走内部升级渠道升级，但是要按格式将资源 copy 到固定路径
[self copyResources];

如果没有相关资源或不需要走 SDK 内部升级，请忽略这一步

```

- (void)copyResources
{
    // 当前版本
    NSDictionary *infoDictionary = [[NSBundle mainBundle] infoDictionary];
    NSString *currentVersion = [infoDictionary objectForKey:@"CFBundleShortVersionString"];

    // 上次copy版本
    NSString *lastVersion = [[NSUserDefaults standardUserDefaults] objectForKey:@"resources_version"];

    // 版本不一致重新copy
    if (![lastVersion isEqualToString:currentVersion]) {

        NSFileManager *fileManager = [NSFileManager defaultManager];

        //工程资源路径
        NSString *sourcePath = [[NSBundle mainBundle] resourcePath];
        NSString *defaultFilePath = [sourcePath stringByAppendingPathComponent:@"demoH5"]; //如果自定义的h5资源文件名称改了，这里的"demoH5"改为新的文件夹名称

        //document存放资源路径
        NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
        NSString *documentsDirectory = [paths objectAtIndex:0];
        // 指定文件夹，用于升级
        NSString *customWebFile = [documentsDirectory stringByAppendingPathComponent:@"customWeb"];
        if (!([fileManager fileExistsAtPath:customWebFile])) {
            [fileManager createDirectoryAtPath:customWebFile withIntermediateDirectories:YES attributes:nil error:nil];
        }

        NSString *writableFilePath = [customWebFile stringByAppendingPathComponent:@"demoH5"]; //如果自定义的h5资源文件名称改了，这里的"demoH5"改为新的文件夹名称

        NSError *error;
        BOOL bfileExist = [fileManager fileExistsAtPath:writableFilePath];
        if (bfileExist) {
            [fileManager removeItemAtPath:writableFilePath error:&error];
        }
        BOOL ret = [fileManager copyItemAtPath:defaultFilePath toPath:writableFilePath error:&error];
        if (ret) {
            [[NSUserDefaults standardUserDefaults] setObject:currentVersion forKey:@"resources_version"];
        }
    }
}

```

4.1.5 消息通知

- 1) 注册消息通知，根据不同的版本调用不同的方法
[self registerRemoteNotification];

```

//注册远程通知
- (void)registerRemoteNotification{
    if (@available(iOS 10.0, *)) { //ios10以上
        //iOS 10
        [[UIApplication sharedApplication] registerForRemoteNotifications];
        [[UNUserNotificationCenter currentNotificationCenter] setDelegate:self];
        UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
        [center requestAuthorizationWithOptions:UNAuthorizationOptionAlert | UNAuthorizationOptionBadge | UNAuthorizationOptionSound completionHandler:^(BOOL granted, NSError * _Nullable error) {
            if (granted) {
            }else {
                //reminder user to open the push notification
            }
        }];
    }else { //ios8-ios10以下
        [[UIApplication sharedApplication] registerForRemoteNotifications];
        [[UIApplication sharedApplication] registerUserNotificationSettings:[UIUserNotificationSettings settingsForTypes:UIUserNotificationTypeAlert | UIUserNotificationTypeBadge | UIUserNotificationTypeSound categories:nil]];
    }
}

```

- 2) 接收系统回调

将系统回调的 deviceToken 通过接口 appRemoteNotificationsWithDeviceToken 设置到 SDK 里面

```

//注册成功 远程通知
- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
    //将deviceToken传入SDK
    [[PbMobileManager shareManager]appRemoteNotificationsWithDeviceToken:deviceToken];
}

//注册失败 远程通知
- (void)application:(UIApplication *)application didFailToRegisterForRemoteNotificationsWithError:(NSError *)error {
}

```

- 3) 接收系统推送消息

针对不同的版本系统回调方法会有不同，建议以下方法都实现，并通过接口 setRemoteNotificationData 传递到 SDK 里面

```
//ios10以下走
- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo {
    [[PbMobileManager shareManager]setRemoteNotificationData(userInfo)];
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler {
    [[PbMobileManager shareManager]setRemoteNotificationData(userInfo)];
}

//ios10及以上
- (void)userNotificationCenter:(UNUserNotificationCenter *)center willPresentNotification:(UNNotification *)notification withCompletionHandler:(void (^)(UNNotificationPresentationOptions options))completionHandler __IOS_AVAILABLE(10.0) __TVOS_AVAILABLE(10.0) __WATCHOS_AVAILABLE(3.0) {
    completionHandler();
    [[PbMobileManager shareManager]setRemoteNotificationData(notification.request.content.userInfo)];
}

- (void)userNotificationCenter:(UNUserNotificationCenter *)center didReceiveNotificationResponse:(UNNotificationResponse *)response withCompletionHandler:(void(^)(void))completionHandler __IOS_AVAILABLE(10.0) __WATCHOS_AVAILABLE(3.0) __TVOS_PROHIBITED {
    completionHandler();
    [[PbMobileManager shareManager]setRemoteNotificationData(response.notification.request.content.userInfo)];
}
```

4.1.6 App 跳转打开

针对分享返回、第三方跳转等，需要在 AppDelegate 里面实现相关系统回调

```
#pragma mark - 跳转第三方App相关
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url {
    return [[PbMobileManager shareManager] appHandleOpenURL:url];
}

- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation {
    return [[PbMobileManager shareManager] appHandleOpenURL:url];
}

- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:(NSDictionary<UIApplicationOpenURLOptionsKey, id> *)options {
    return [[PbMobileManager shareManager] appHandleOpenURL:url];
}

- (BOOL)application:(UIApplication *)application continueUserActivity:(NSUserActivity *)userActivity restorationHandler:(void(^)(NSArray<id<UIUserActivityRestoring> * __nullable restorableObjects))restorationHandler {
    return [[PbMobileManager shareManager] handleOpenUniversalLinkWithUserActivity:userActivity];
}
```

4.1.7 屏幕旋转

针对详情页横屏功能，需要设置系统旋转方法

```
#pragma mark - Portrait And Landscape
//获取支持屏幕旋转的方向
- (UIInterfaceOrientationMask)application:(UIApplication *)application supportedInterfaceOrientationsForWindow:(UIWindow *)window {
{
    return [[PbMobileManager shareManager] getSupportedInterfaceOrientationsForWindow>window];
}

// 获取是否支持自动旋转屏幕
- (BOOL)shouldAutorotate {
    return [[PbMobileManager shareManager] getSupportAutorotateWithWindow:self.window];
}
```

4.2 方案二（单页面模式）使用流程

重要提示：

1、SDK 启动相关代码都在 AppDelegate 中添加，可参考 Demo 工程，建议直接拷贝相关代码使

用。如果接入方是使用 XCode11 以上版本创建的新项目，需要删除 SceneDelegate（删除方法见备注 7.1），或者把 window 创建等相关 UI 代码转移到 SceneDelegate 中。

2、APP 主导航必须基于 UINavigationController 创建，否则会影响 SDK 内部的页面打开和跳转逻辑。

4. 2. 1 在 APP 启动时初始化 SDK

在 APP 入口 AppDelegate 中加入初始化代码，根据需要 import 相关头文件

//添加 window 和主导航栏的声明

```
@property (strong, nonatomic) UIWindow *window;
```

```
@interface AppDelegate ()<UNUserNotificationCenterDelegate,PbMobileSDKDelegate>
{
    UITabBarController *mTabBar;

    UIView *mLoadingView;
    UIProgressView *mProgressView; // 进度条
}

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    [self.window makeKeyAndVisible];

    //1.获取PbMobileManager单例
    PbMobileSDK *sdk = [PbMobileSDK shareMobileSDK];
    //2.设置SDK状态回调代理
    [sdk setDelegate:self];
    //3.启动SDK
    [sdk startSDKWithOptions:launchOptions];
    //4.创建tabbar, Navigation
    mTabBar = [[UITabBarController alloc] init];
    mTabBar.tabBar.translucent = NO; //关闭透明效果
    //5.创建UINavigationController并指定根控制器
    UINavigationController *navCtr = [[UINavigationController alloc] initWithRootViewController:mTabBar];
    //6.隐藏navigationBar
    navCtr.navigationBar.hidden = YES;
    //7.设置window的跟控制器为 UINavigationController
    self.window.rootViewController = navCtr;
    //8.设置Navigation到SDK里面
    [sdk setRootNavigationController:navCtr];
    //9.注册消息通知
    [self registerRemoteNotification];
    //10.设置交易产品标识
    [[PbMobileDataManager shareDataManager] setTradeLoginProduct:25];

    //添加启动页（可实现启动进度展示，按需实现，也可以没有）
    [self addLoadingView];

    return YES;
}
```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)la
unchOptions

```
self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
```

```
[self.window makeKeyAndVisible];
```

```

//1.获取 PbMobileManager 单例
PbMobileSDK *sdk = [PbMobileSDK shareMobileSDK];
//2.设置 SDK 状态回调代理
[sdk setDelegate:self];
//3.启动 SDK
[sdk startSDKWithOptions:launchOptions];
//4.创建 tabBar, Navigation
mTabBar = [[UITabBarController alloc] init];
mTabBar.tabBar.translucent = NO; //关闭透明效果
//5.创建 UINavigationController 并指定根控制器
UINavigationController *navCtr = [[UINavigationController alloc] initWithRootViewController:mTa
bBar];
//6.隐藏 navigationBar
navCtr.navigationBar.hidden = YES;
//7.设置 window 的跟控制器为 UINavigationController
self.window.rootViewController = navCtr;
//8.设置 Navigation 到 SDK 里面
[sdk setRootNavigationController:navCtr];
//9.注册消息通知
[self registerRemoteNotification];
//10.设置交易产品标识 (产品标识 SDK 目前只能设置为 25, 如需更改请咨询澎博) (产品标
识设置 25; 柜台添加: POBO_APPSDK_1.0.0 这个 APPID, 澄博的交易网关修改 APPID 新
增 POBO_APPSDK_1.0.0 APPID 配置项)
[[PbMobileDataManager shareDataManager] setTradeLoginProduct:25];

//添加启动页 (可实现启动进度展示, 按需实现, 也可以没有)
// [self addLoadingView];

```

4. 2. 2 实现状态回调代理协议接口

在 APP 入口 AppDelegate 中实现状态回调代理协议接口

```

#pragma mark - 澎博SDK状态回调代理协议
// 澎博SDK状态回调代理协议，可根据SDK启动状态来实现启动页或进度展示
- (void)PbMobileSDKStartProcessState:(PbStartProcess)state
{
    switch (state) {
        case PbProcessState_DownloadNative:          //批量下载
            [mProgressView setProgress:0.1];
            break;
        case PbProcessState_UpdatePbres_Start:      //pbres资源升级开始
            [mProgressView setProgress:0.2];
            break;
        case PbProcessState_UpdatePbres_Finish:     //pbres升级结束
            [mProgressView setProgress:0.4];
            break;
        case PbProcessState_HQ_Start:                //行情开始登陆
            [mProgressView setProgress:0.5];
            break;
        case PbProcessState_HQ_DownloadMarket:       //行情开始下载市场文件
            [mProgressView setProgress:0.6];
            break;
        case PbProcessState_HQ_DownloadCode:         //行情开始下载码表文件
            [mProgressView setProgress:0.7];
            break;
        case PbProcessState_HQ_Ready:                 //行情服务就绪
            [mProgressView setProgress:0.9];
            break;
        case PbProcessState_Success:                //启动完成
        {
            [mProgressView setProgress:1];
            //创建导航栏页面：自选、行情、交易和其他自定义页面
            [self createTabBarViewControllers];

            //移除启动页
            [mLoadingView removeFromSuperview];
        }
        break;
    default:
        break;
    }
}

```

详细代码参考 Demo 工程

4.2.3 SDK 启动完成，创建主导航页面：行情、交易等

在 SDK 启动完成后，创建主导航页面：自选、行情、交易和其它自定义页面。
自选、行情、交易页面提供 ViewController 供接入方 APP 嵌入导航栏。

期货			
主力合约	中金所	上期所	大商所
名称	最新价	涨跌	仓差
沪铜1803 cu1803	52710	-930	22136
沪铝1803 al1803	14645	-20	-8072
沪锌1803 zn1803	25910	-215	4628
沪铅1803 pb1803	19595	15	-1676
黄金1806 au1806	280.40	1.30	6282
白银1806 ag1806	3834	-9	23840
螺纹1805 rb1805	3939	34	140438
橡胶1805 ru1805	13780	-140	-2818
沥青1806 bu1806	2854	78	127538
	首页	自选	行情
			交易
			我的

```
//创建导航栏页面：自选、行情、交易和其他自定义页面
- (void)createTabBarViewControllers
{
    //创建自定义首页
    PbDemoHomeViewController *homeVC = [[PbDemoHomeViewController alloc] init];
    homeVC.tabBarItem.image = [[UIImage imageNamed:@"home1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
    homeVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"home2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];

    //创建自选页面
    PbSelfStockViewController *pbSelfStockVC = [[PbSelfStockViewController alloc] initWithLoadedInTabBar:YES];
    [pbSelfStockVC addCustomTilteView:@""];
    pbSelfStockVC.tabBarItem.image = [[UIImage imageNamed:@"zixuan1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
    pbSelfStockVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"zixuan2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];

    //创建行情页面
    PbHQViewController *pbHQVC = [[PbHQViewController alloc] initWithLoadedInTabBar:YES];
    [pbHQVC addCustomTilteView:@""];
    pbHQVC.tabBarItem.image = [[UIImage imageNamed:@"hq1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
    pbHQVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"hq2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];

    //创建交易页面
    PbTradeViewController *pbTradeVC = [[PbTradeViewController alloc] initWithLoadedInTabBar:YES];
    [pbTradeVC addCustomTilteView:@""];
    pbTradeVC.tabBarItem.image = [[UIImage imageNamed:@"trade1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
    pbTradeVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"trade2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];

    //创建自定义页面
    PbDemoMineViewController *mineVC = [[PbDemoMineViewController alloc] init];
    mineVC.tabBarItem.image = [[UIImage imageNamed:@"mine1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
    mineVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"mine2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];

    //主导航初始化页面控制器
    mTabBar.viewControllers = @[homeVC, pbSelfStockVC, pbHQVC, pbTradeVC, mineVC];
}

//创建自定义首页
PbDemoHomeViewController *homeVC = [[PbDemoHomeViewController alloc] init];
homeVC.tabBarItem.image = [[UIImage imageNamed:@"home1.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
homeVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"home2.png"] imageWithRenderingMode:UIImageRenderingModeAlwaysOriginal];
```

```

//创建自选页面
PbSelfStockViewController *pbSelfStockVC = [[PbSelfStockViewController alloc] initWithLoadedIn
TabBar:YES];
[pbSelfStockVC addCustomTilteView:@""];
pbSelfStockVC.tabBarItem.image = [[UIImage imageNamed:@"zixuan1.png"] imageWithRenderin
gMode: UIImageRenderingModeAlwaysOriginal];
pbSelfStockVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"zixuan2.png"] imageWithR
enderingMode: UIImageRenderingModeAlwaysOriginal];

//创建行情页面
PbHQViewController *pbHQVC = [[PbHQViewController alloc] initWithLoadedInTabBar:YES];
[pbHQVC addCustomTilteView:@""];
pbHQVC.tabBarItem.image = [[UIImage imageNamed:@"hq1.png"] imageWithRenderingMode:UI
ImageRenderingModeAlwaysOriginal];
pbHQVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"hq2.png"] imageWithRendering
Mode: UIImageRenderingModeAlwaysOriginal];

//创建交易页面
PbTradeViewController *pbTradeVC = [[PbTradeViewController alloc] initWithLoadedInTabBar:YES
];
[pbTradeVC addCustomTilteView:@""];
pbTradeVC.tabBarItem.image = [[UIImage imageNamed:@"trade1.png"] imageWithRenderingMo
de: UIImageRenderingModeAlwaysOriginal];
pbTradeVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"trade2.png"] imageWithRend
eringMode: UIImageRenderingModeAlwaysOriginal];

//创建自定义页面
PbDemoMineViewController *mineVC = [[PbDemoMineViewController alloc] init];
mineVC.tabBarItem.image = [[UIImage imageNamed:@"mine1.png"] imageWithRenderingMode:
UIImageRenderingModeAlwaysOriginal];
mineVC.tabBarItem.selectedImage = [[UIImage imageNamed:@"mine2.png"] imageWithRenderin
gMode: UIImageRenderingModeAlwaysOriginal];
//主导航初始化页面控制器
mTabBar.viewControllers = @[homeVC, pbSelfStockVC, pbHQVC, pbTradeVC, mineVC];

```

4. 2. 4 注册和接收消息通知

在 APP 入口 AppDelegate 中实现消息通知相关接口

```

#pragma mark - 消息通知相关

//注册远程通知
- (void)registerRemoteNotification{
    [[UIApplication sharedApplication] registerForRemoteNotifications];
    if (@available(iOS 10.0, *)) { //ios10以上
        //iOS 10
        [[UNUserNotificationCenter currentNotificationCenter] setDelegate:self];
        UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
        [center requestAuthorizationWithOptions:UNAuthorizationOptionAlert | UNAuthorizationOptionBadge |
         UNAuthorizationOptionSound completionHandler:^(BOOL granted, NSError * _Nullable error) {
            if (granted) {
            }else {
                //reminder user to open the push notification
            }
        }];
    }else { //ios10以下

        [[UIApplication sharedApplication] registerForRemoteNotifications];
        [[UIApplication sharedApplication] registerUserNotificationSettings:[UIUserNotificationSettings
            settingsForTypes:UIUserNotificationTypeAlert | UIUserNotificationTypeBadge | UIUserNotificationTypeSound
            categories:nil]];
    }
}

//注册成功 远程通知
- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
    //将deviceToken传入SDK
    [[PbMobileSDK shareMobileSDK]appRemoteNotificationsWithDeviceToken:deviceToken];
}

//注册失败 远程通知
- (void)application:(UIApplication *)application didFailToRegisterForRemoteNotificationsWithError:(NSError *)error {

}

//ios10以下走
- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo {
    [[PbMobileSDK shareMobileSDK]setRemoteNotificationData:userInfo];
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo
    fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler {
    [[PbMobileSDK shareMobileSDK]setRemoteNotificationData:userInfo];
}

//iOS10及以上
- (void)userNotificationCenter:(UNUserNotificationCenter *)center willPresentNotification:(UNNotification *)notification
    withCompletionHandler:(void (^)(UNNotificationPresentationOptions options))completionHandler __IOS_AVAILABLE(10.0)
    __TVOS_AVAILABLE(10.0) __WATCHOS_AVAILABLE(3.0) {
    completionHandler(0);
    [[PbMobileSDK shareMobileSDK]setRemoteNotificationData:notification.request.content.userInfo];
}

- (void)userNotificationCenter:(UNUserNotificationCenter *)center didReceiveNotificationResponse:(UNNotificationResponse *)
    *response withCompletionHandler:(void(^)(void))completionHandler __IOS_AVAILABLE(10.0) __WATCHOS_AVAILABLE(3.0)
    __TVOS_PROHIBITED {
    completionHandler();
    [[PbMobileSDK shareMobileSDK]setRemoteNotificationData:response.notification.request.content.userInfo];
}

```

详细代码参考 Demo 工程

4.2.5 SDK 主题(黑白皮肤)切换回调

```
#pragma mark - 澎博SDK主题切换时回调主题ID
/// 主题改变时的回调方法，以供SDK外部根据不同的主题适配自定义页面
/// @param themeId 主题ID，目前支持两种，@"Theme_White":白色主题 @"Theme_Black":黑色主题
- (void)themeChangedWithThemeId:(NSString *)themeId{
    NSLog(@"当前的主题ID: %@", themeId);
}
```

详细代码参考 Demo 工程

4.2.6 自动横竖屏控制

```
#pragma mark - Portrait And LandScape
//获取支持屏幕旋转的方向
- (UIInterfaceOrientationMask)application:(UIApplication *)application supportedInterfaceOrientationsForWindow:(UIWindow *)window
{
    return [[PbMobileSDK shareMobileSDK] getSupportedInterfaceOrientationsForWindow:window];
}

// 获取是否支持自动旋转屏幕
- (BOOL)shouldAutorotate
{
    return [[PbMobileSDK shareMobileSDK] getSupportAutorotateWithWindow:self.window];
}
```

详细代码参考 Demo 工程

4.2.7 分享相关处理

```
#pragma mark - 分享之后从第三方App跳转返回
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url {
    return [[PbMobileSDK shareMobileSDK] appHandleOpenURL:url];
}

- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication
annotation:(id)annotation {
    return [[PbMobileSDK shareMobileSDK] appHandleOpenURL:url];
}

- (BOOL)application:(UIApplication *)application continueUserActivity:(NSUserActivity *)userActivity
restorationHandler:(void(^)(NSArray<id<UIUserActivityRestoring>> * __nullable restorableObjects))restorationHandler {
    return [[PbMobileSDK shareMobileSDK] handleOpenUniversalLinkWithUserActivity:userActivity];
}
```

详细代码参考 Demo 工程

4.2.8 云认证接口

如果接入方需要自己实现云认证登录页，在此处打开自己的云认证登录页面，在登录成功之后调用相关接口把信息传给 SDK 即可。如果直接使用澎博云认证请保持原样。

```
#pragma mark - PbMobileManagerDelegate
// 打开认证页面
- (void)loadAppCertifyView:(UIViewController *)sender
{
    [[PbMobileSDK shareMobileSDK] openViewController:sender withUrl:@"pobo:pageId=800001"];
}
```

详细代码参考 Demo 工程

4.3 澄博 SDK 接口使用

详细请查看《澄博手机 SDK 接口文档_iOS.docx》

5 澎博 SDK 中涉及第三方 SDK 集成说明

澎博 SDK 中有集成相关功能的第三方 SDK，比如百度统计 SDK、微信分享 SDK、极光推送 SDK 等等，APP 接入时可根据功能需要按需集成。

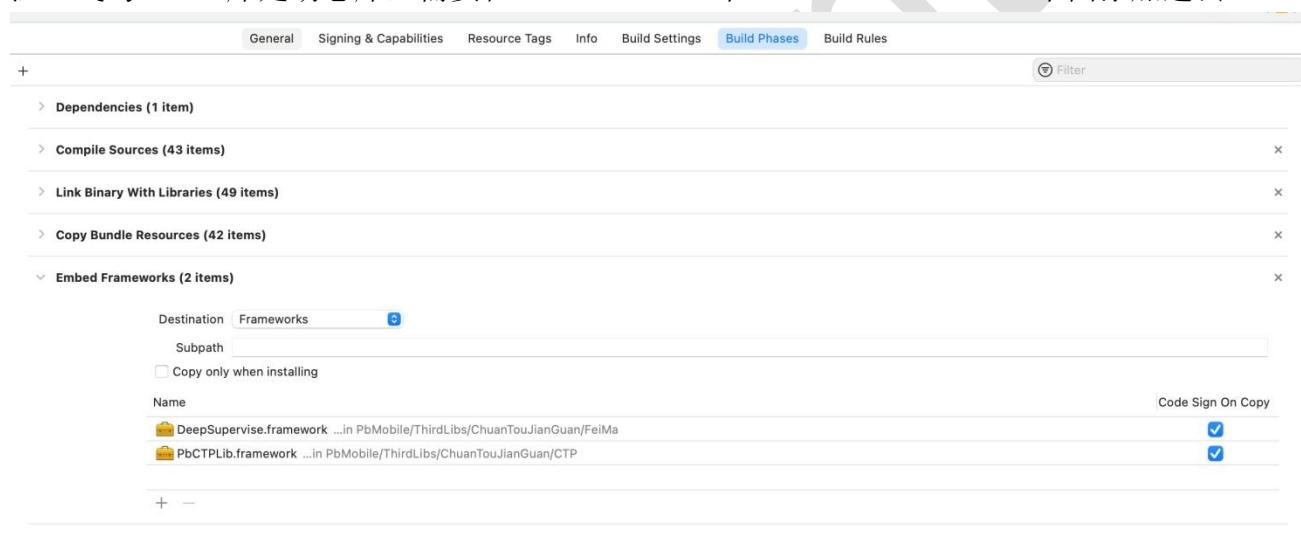
相关 SDK 库全部在 ThirdLibs 文件夹下面，如果不需要，可直接删除。

5.1 穿透式监管 SDK

目录在 ThirdLibs->ChuanTouJianGuan 文件下面，根据交易站点配置的柜台属性选择加入，将多余的直接删除

—— FeiMa	飞马穿透式库（动态库 DeepSupervise.framework）
—— CTP	CTP 穿透式库（动态库 PbCTPLib.framework）
—— JinShiDa	金仕达穿透式库
—— AXSystemInfoLib	顶点穿透式库
—— HsInfoCollect	恒生穿透式库

注：飞马、CTP 库是动态库，需要在 Build Phase 下 Embed Frameworks 下面添加进去



5.2 分享 SDK（微信、QQ）

目录在 ThirdLibs->ShareSDK 文件下面，根据支持的分享可以动态选择，不需要的可以直接删除，文件夹下面包含：

1) PbShareLib.framework

澎博分享 SDK 合集，有分享必须包含，里面处理分享相关库的动态加载

2) OpenSDK

微信分享 SDK，支持微信分享需到微信开放平台申请相关 ID（存在审核时间，时间由微信控制）

根据申请到的 ID 进行相关配置：

分享的 appID、universalLink 博易 APP 配置中心后台配置修改；

Info->URL Types 添加配置

The screenshot shows the Xcode project settings for the 'Info' tab. Under 'URL Types (4)', there is an entry for 'weixin'. The configuration includes:

- Identifier:** weixin
- Icon:** None
- URL Schemes:** wxd8223a84bbf41872
- Role:** Editor

Info.plist 文件下面 LSApplicationQueriesSchemes 添加微信相关, 实现微信分享相关的 APP 间的跳转

LSApplicationQueriesSchemes			Array	(31 items)
Item 0			String	weixinULAPI
Item 1			String	weixin

3) TencentOpenAPI.framework

QQ 分享 SDK, 支持 QQ 分享需到腾讯开放平台申请相关 ID

根据申请到的 ID 进行相关配置:

分享的 appID、universalLink 博易 APP 配置中心后台配置修改;

Info->URL Types 添加配置

The screenshot shows the Xcode project settings for the 'Info' tab. Under 'URL Types (4)', there is an entry for 'qq'. The configuration includes:

- Identifier:** qq
- Icon:** None
- URL Schemes:** tencent101775611
- Role:** Editor

Info.plist 文件下面 LSApplicationQueriesSchemes 添加微信相关, 实现 QQ 分享相关的 APP 间的跳转

Bundle version	String	\$(CURRENT_PROJECT_VERSION)
LSApplicationQueriesSchemes	Array	(27 items)
Item 0	String	weixinULAPI
Item 1	String	weixin
Item 2	String	mqq
Item 3	String	mqqapi
Item 4	String	mqqOpensdkSSoLogin
Item 5	String	mqqconnect
Item 6	String	mqqopensdkdataline
Item 7	String	mqqopensdkgrouptribeshare
Item 8	String	mqqopensdkfriend
Item 9	String	mqqopensdkapi
Item 10	String	mqqopensdkapiV2
Item 11	String	mqqopensdkapiV3
Item 12	String	mqzoneopensdk
Item 13	String	wtloginmqq
Item 14	String	wtloginmqq2
Item 15	String	mqqwpa
Item 16	String	mqzone
Item 17	String	mqzonev2
Item 18	String	mqzoneshare
Item 19	String	wtloginqzone
Item 20	String	mqzonewx
Item 21	String	mqzoneopensdkapiV2
Item 22	String	mqzoneopensdkapi19
Item 23	String	mqzoneopensdkapi
Item 24	String	mqzoneopensdk
Item 25	String	alipay
Item 26	String	alipayshare

5.3 百度统计 SDK

目录在 ThirdLibs->BaiduMobStat，如需使用百度统计，将该文件夹放入工程，并从百度开发者网站申请相关 ID，百度统计申请的 ID 在博易 APP 配置中心后台配置修改。

5.4 云屋 SDK

目录在 ThirdLibs-> VideoCall，如需使用云屋功能，需要申请相关信息，然后将对应的服务器、账号密码等在博易 APP 配置中心后台配置

5.5 极光一键登陆

目录在 ThirdLibs-> TelecomLogin，将该文件放入工程即可，在极光相关网站申请开通认证功能，并将申请的极光相关的 key，配置到博易 APP 配置中心后台。

5.6 通付盾 OCR SDK

目录在 ThirdLibs-> OCRSRK，如需使用通付盾功能，需要申请相关信息，然后将对应的 appKey、aesKey 等在博易 APP 配置中心后台配置，且根据 bundID 会申请到一个唯一的 CWResource.bundle，也要加上到工程里面。

5.7 证通 OCR SDK

目录在 ThirdLibs-> ZTOCRSDK，如需使用证通功能，需要申请相关信息，然后将对应的信息在博易 APP 配置中心后台配置。

5.8 天威云 OCR SDK

目录在 ThirdLibs-> TWOCRSDK，如需使用天威云功能，需要申请相关信息，
PbThirdSDKConfig.cfg 配置

[TWOCR]

appId=

secretId=

serverIp=

6 附录 (Appendix)

6.1 SDK 接口定义

请参见《澎博 iOS SDK 接口文档 V1.0. docx》(当前版本 V1.0)

6.2 SDK 功能相关配置定义

请参见《博易 APP V5.5 原生配置说明文档_iOS. docx》(当前版本 V5.5)

6.3 博易 APP 使用说明

请参见《博易 APP V5.5 使用说明书 20210317. pdf》(当前版本 V5.5)

6.4 博易 APP 测试报告

请参见《博易 APP V5.5 测试报告-IOS. pdf》(当前版本 V5.5)

7 备注

7.1 删 除 XCode11 新建项目中的 SceneDelegate

Xcode11之后新创建的工程会多出两个文件SceneDelegate。那么我们如何让它变回之前的那样的工程呢。

- 一、将这两个文件删除。

会报错 `There is no scene delegate set. A scene delegate class must be specified to use a main storyboard`

- 二、将Info.plist -> Open As -> Source Code将划框的部分删除。

```

20      <string>i</string>
21      <key>LSRequiresiPhoneOS</key>
22      <true/>
23      <key>UIApplicationSceneManifest</key>
24      <dict>
25          <key>UIApplicationSupportsMultipleScenes</key>
26          <false/>
27          <key>UISceneConfigurations</key>
28          <dict>
29              <key>UIWindowSceneSessionRoleApplication</key>
30              <array>
31                  <dict>
32                      <key>UISceneConfigurationName</key>
33                      <string>Default Configuration</string>
34                      <key>UISceneDelegateClassName</key>
35                      <string>SceneDelegate</string>
36                      <key>UISceneStoryboardFile</key>
37                      <string>Main</string>
38                  </dict>
39              </array>
40          </dict>
41      </dict>
42      <key>UILaunchStoryboardName</key>
43      <string>LaunchScreen</string>
44      <key>UIMainStoryboardFile</key>
```

- 三、将 `AppDelegate.m` 中的 `UISceneSession lifecycle` 注释掉或者删掉。

```

///#pragma mark - UISceneSession lifecycle
//
//-( UISceneConfiguration *)application:(UIApplication *)application
//    configurationForConnectingSceneSession:(UISceneSession *)connectingSceneSession options:(UISceneConnectionOptions *)
//    *options {
//    // Called when a new scene session is being created.
//    // Use this method to select a configuration to create the new scene with.
//    // return [[UISceneConfiguration alloc] initWithName:@"Default Configuration" sessionRole:connectingSceneSession.role]
//}
//
//-( void)application:(UIApplication *)application didDiscardSceneSessions:(NSSet<UISceneSession *> *)sceneSessions {
//    // Called when the user discards a scene session.
//    // If any sessions were discarded while the application was not running, this will be called shortly after
//    application:didFinishLaunchingWithOptions.
//    // Use this method to release any resources that were specific to the discarded scenes, as they will not return.
//}
```