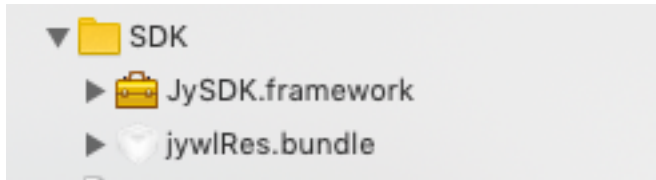


说明文档

一、SDK结构

1. SDK核心框架和资源。



支持平台: iPod Touch,iPhone,iPad。 系统要求: iOS7.0+,

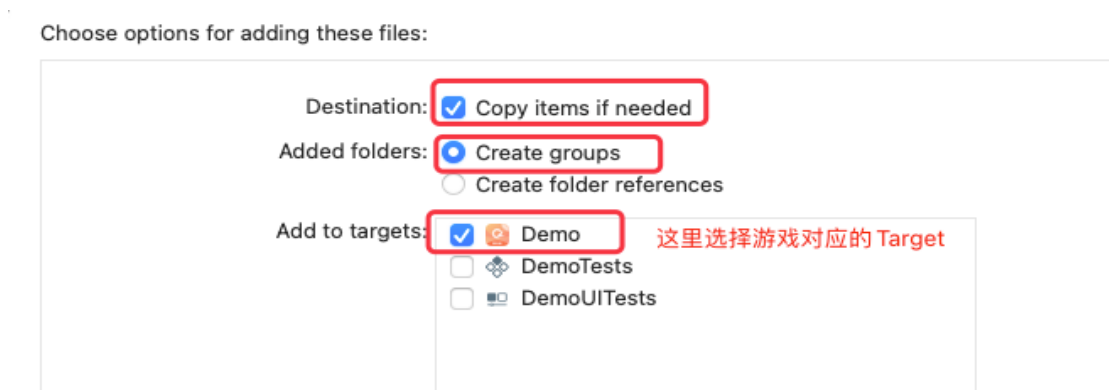
支持框架: armv7 arm64

环境要求: xcode11.0+

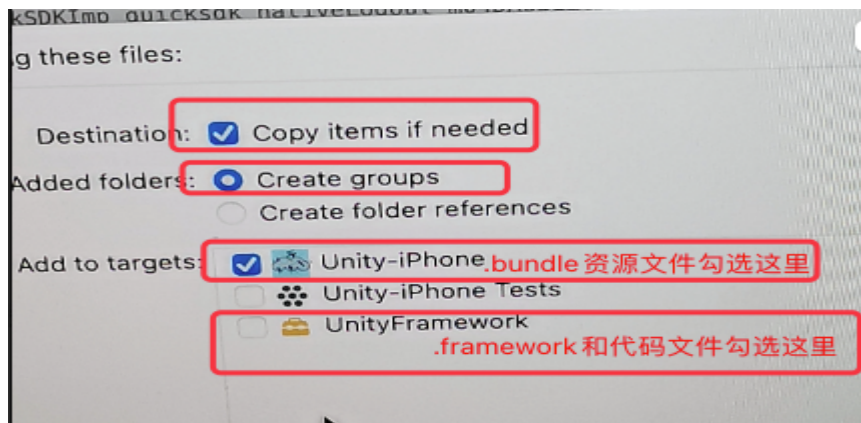
2.SDK本身支持全方向

在游戏内SDK具体的展示方向由游戏的delegate.window.rootviewController来决定。

3.添加SDK文件夹下的所有文件到工程中，并选择游戏所在的Target,如下图所示：



注意：如果游戏为Unity项目，针对使用Unity version 2019.3之后的版本导出的xcode工程，由于工程中新增了Target: UnityFramework，导入SDK文件时，需要将SDK中所有的.framework 和.a文件添加到UnityFramework这个Target；只需要将.bundle图片等资源文件添加到游戏对应的Target。如下图所示：



二、注意事项

游戏内支持购买自动订阅型商品或者需要检测苹果退款时，需要在苹果后台配置服务器通知网址，URL格式：**xxx/notify/apple**。其中xxx为厂商的sdk完整域名，必须带https。配置时需选择版本1示例如下图：



另外，游戏内支持购买自动订阅型商品还需要在quickgame后台产品详情里配置苹果后台的共享密钥，



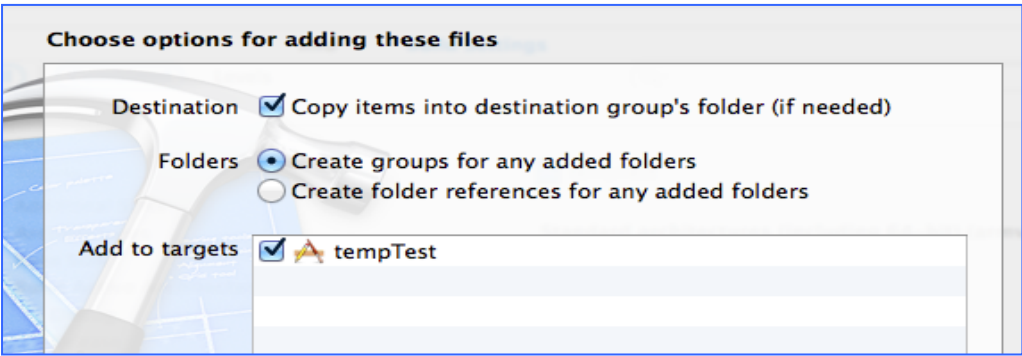
获取苹果后台生成共享密钥的入口：



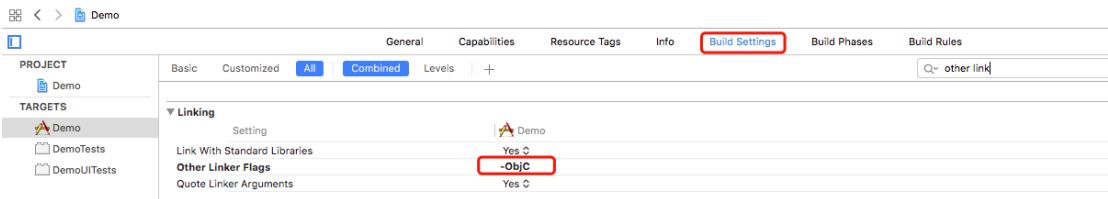
三、接入步骤

特别说明:请参照接入文档接入，如有疑问可参考demo，最终以demo为准。

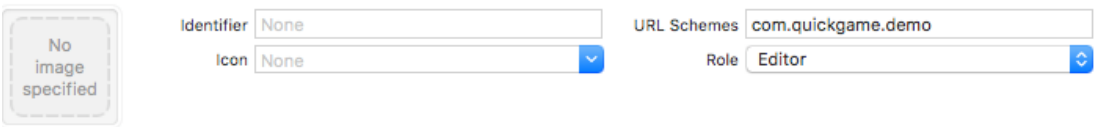
添加SDK，将SDK文件夹导入xcode工程中；选择正确的target



在Build Settings->Linking->Other Linking Flags 中配置 -ObjC



将应用的Bundle Identifier添加到URL scheme,



配置允许访问Http网络

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

SDK需要申请的权限说明

SDK默认只需要添加相册权限用于保存游客账号信息，info.plist 配置以下内容

Privacy - Photo Library Additions Usage Description	String	亲需要使用你的相册来保存游客账号信息
Privacy - Photo Library Usage Description	String	亲需要使用你的相册来保存游客账号信息

如果需要使用QuickAD, TapDB, Appsflyer, 热云广告统计，还需要添加ATT权限用于获取idfa

▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Privacy - Photo Library Additions Usage Description	String	亲需要使用你的相册来保存游客账号信息
Privacy - Photo Library Usage Description	String	亲需要使用你的相册来保存游客账号信息
Privacy - Camera Usage Description	String	亲需要使用你的相机来给客服发送图片信息
Privacy - Tracking Usage Description	String	获取 idfa 用于广告归因

如果带有Quick云客服系统，需要添加相机和相册权限用于给客服人员发送图片信息

Privacy - Photo Library Additions Usage Description	String	亲需要使用你的相册来保存游客账号信息
Privacy - Photo Library Usage Description	String	亲需要使用你的相册来保存游客账号信息
Privacy - Camera Usage Description	String	亲需要使用你的相机来给客服发送图片信息
Privacy - Tracking Usage Description	String	获取 idfa 用于广告归因

需要接入热云SDK时添加如下系统库：

AdSupport.framework、CoreMotion.framework、iAd.framework、libresolv.9.tbd、libresolv.tbd、AdServices.framework(optional形式引入)

接入说明：可参考sdk附带的demo和本文档进行接入

四、接口说明

1. SDK初始化（必接）

```
/**
 * @brief SDK平台初始化方法（必接）
 * @param productCode 产品code
 * @param completion 初始化结果回调
 */
+ (void) initWithProductCode: (NSString *)productCode completion:
(KInitResultBlock) completion;
```

功能：使用产品id，初始化SDK。参数：productCode,产品id，必填，对接商务提供。

初始化结果通过block回传

代码实例：

```
[JySDKManager initWithProductCode:SDK_PRODUCT_CODE completion:^(Status_CODE
retCode) {
    if (retCode == kInitSuccess) {
        NSLog(@"初始化成功");
    } else {
        NSLog(@"初始化失败，错误码: %d", retCode);
    }
}];
```

2. 展示用户协议和隐私政策弹窗（选接）

/** 显示用户隐私弹窗， 用户同意返回YES，不同意返回NO*/

+ (void)showPrivacyViewController:(kPrivacyBlock)isAgreed;

功能:展示SDK自带的用户协议和隐私政策，一般在登录之前调用，用户同意协议后调用登录。

3. QuickAD接入（选接）

在- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions调用

[JySDKManager initQkAd:@"你的quickad参数"];

参数传入productcode在quick后台获取，AD仅需要初始化，SDK内部会自行实现统计。

4. 登录接口(必接)

```
/**
 * @brief 调起登录界面（必接）
 * @param success 登录成功结果回调，返回用户信息
 * @param failure 登录失败结果回调
 */
+ (void)loginWithSuccBlock:(KLoginResultBlock)succ failBlock:
(KLoginFailResult)failure;
```

功能：进入用户登录页面。需要在初始化成功之后调用，否则调用无效。游戏开始时可以调用此接口显示一个用户登录界面。在收到用户注销回调后通常处理办法是回到游戏登录界面调用后调用此接口。

实例代码：

```
[JySDKManager loginWithSuccBlock:^(NSDictionary *resultDic) {
    NSString *code = [resultDic objectForKey:@"code"];
    NSString *token = [JySDKManager userToken];
    switch (code.integerValue) {
        case kErrorNone:{
            NSString *userid = [resultDic objectForKey:@"userId"];
            NSLog(@"登录成功:\n用户ID:%@, 验证码:%@",userid,token);

            if([JySDKManager isGuest]){
                NSLog(@"是游客登录");
            }
        }
    }
}];
```

```

        }else{
            NSLog(@"非游客登录");
        }
        // 登录成功, 上传角色信息
        GameRole *role = [GameRole new];
        role.roleId = @"testRoleid"; /// 必传
        role.role_name = @"testRoleName";
        role.serverId = @"1";
        role.sv_name = @"testServer";
        role.role_level = @"testRoleLevel";
        role.vipLevel = @"testVipLevel";
        [JySDKManager updateRoleInfo:role];
        NSString *age = JySDKManager.getCurrentUserAge;
        NSLog(@"当前用户UID=%@, 年龄=%@", JySDKManager.userId, age);
    }

    break;

default:
    break;
}
} failBlock:^(NSString *message) {
    NSLog(@"登录失败: %@", message);
}];

```

5. 充值（必接）

```

/**
 * @brief 购买接口（必接）
 *
 * @param param 充值参数
 * @param completion 购买结果回调
 */
+ (void)getGoodWithParam:(GoodParam *)param completion:
(KPyResultBlock)completion;

```

功能：调起苹果内购。

参数：param 充值信息,参考GoodParam.h文件 。注意：其中扩展参数请勿传特殊符号，如果无法避免建议先进行base64编码后再传

说明：异步充值，sdk通知的结果仅供参考，以服务器端同步为准。

6. 静默登录接口（选接）

```

/** 静默登录，不显示登录界面
 * @param isShowFloatMenu 是否显示浮标
 * @param success 登录成功回调,包含用户相关信息
 * @param failure 登录失败回调
 */
+ (void)loginAsGuestWithFloatMenuShow:(BOOL)isShowFloatMenu successBlock:
(KLoginResultBlock)success failBlock:(KLoginFailResult)failure;

```

功能：静默登录SDK，不显示登录界面。需要在初始化成功之后调用，否则调用无效；

参数：isShowFloatMenu通过传入参数YES或者NO控制是否显示浮标

failure：登录失败回调，回调参数为字符串类型为失败原因

代码调用示例:

```
[JySDKManager loginAsGuestWithFloatMenuShow:YES successBlock:^(NSDictionary
*resultDic) {
    NSLog(@"静默登录成功, 返回用户信息: %@",resultDic);
    // 状态码 1000代表登录成功; 各状态码含义请查看JySDKManager.h
    NSString *code = [resultDic objectForKey:@"code"];
    // 用户唯一id
    NSString *userId = [resultDic objectForKey:@"userId"];
    //userToken用户校验码
    NSString *token = [JySDKManager userToken];
    // 用户名
    NSString *account = [resultDic objectForKey:@"account"];
    // 登录方式
    NSString *loginType = [resultDic objectForKey:@"loginType"];
    // 登录成功附带消息
    NSString *message = [resultDic objectForKey:@"message"];

} failBlock:^(NSString *message) {
    NSLog(@"静默登录失败, 原因: %@",message);
}];
```

7. 微信登录 (选接)

appdelegate必须实现以下三个方法, 可参考demo

```
/** 微信QQ授权登录回调方法, 带第三方登录必接*/
+ (BOOL)application:(UIApplication *)application continueUserActivity:
(NSUserActivity *)userActivity restorationHandler:(void (^)(
NSArray<id<UIUserActivityRestoring>> * _Nullable))restorationHandler;
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
options:(NSDictionary *)options;
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)source annotation:(id)annotation;
```

info.plist添加微信白名单, 没有LSApplicationQueriesSchemes就添加一个

▼ LSApplicationQueriesSchemes	Array	(4 items)
Item 0	String	weixin
Item 1	String	weixinULAPI

新版微信需要, 2020年4月后接入的需要

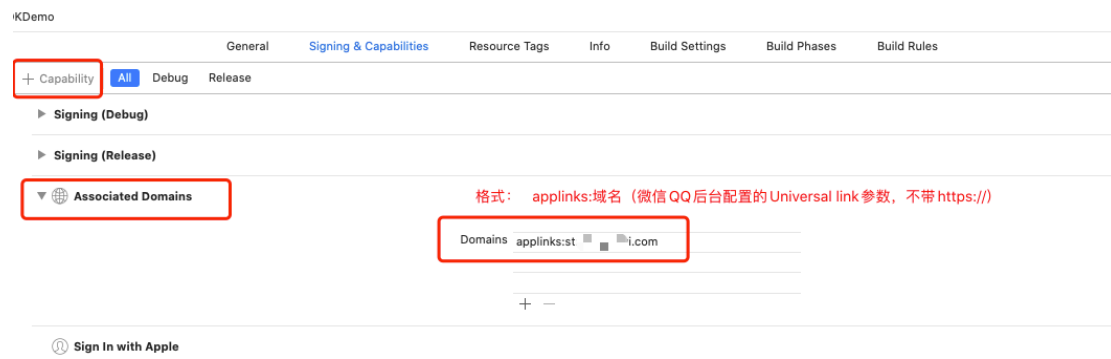
微信对Universal Links配置要求

a) Universal Links必须支持https

b) Universal Links配置的paths不能带query参数

c) 微信使用Universal Links拉起第三方App时, 会在Universal Links末尾拼接路径和参数, 因此App配置的paths必须加上通配符/*

在Signing & Capabilities添加Associated Domains，内容：applinks:域名，注意：这里域名不带https。



将微信appid参数添加到info.plist文件的URL scheme



初始化微信SDK在- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions调用

调用示例：

```
[JySDKManager initWithWxLogin:@"wxbacbb624a09218cd"  
appSecret:@"55e28c7aa650cad991ab918371bab7b4" universalLink:@"https://  
dl.sxyxserver.com/mqyy/apple-app-site-association/"];
```

8. 微信分享结果回调-选接(也可自行转接)

```
[JySDKManager wxShareResult:^(NSDictionary * _Nonnull resultDic) {  
    NSString * msg = [resultDic objectForKey:@"msg"];  
    if ([msg isEqualToString:@"succ"]) {  
        NSLog(@"分享成功");  
    }  
    else if ([msg isEqualToString:@"cancel"]) {  
        NSLog(@"用户取消分享");  
    } else {  
        NSLog(@"分享失败");  
    }  
}];
```

返回参数类型:字典

键:msg

值:succ 分享成功

fail 分享失败

cancel 用户取消分享

9. QQ登录（选接）

AppDelegate必须接入以下方法

```
+ (BOOL)application:(UIApplication *)application continueUserActivity:
(NSUserActivity *)userActivity restorationHandler:(void (^)(
NSArray<id<UIUserActivityRestoring>> * _Nullable))restorationHandler;
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
options:(NSDictionary *)options;
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)source annotation:(id)annotation;
```

info.plist添加QQ白名单 –info.plist没有LSApplicationQueriesSchemes就添加一个

▼ LSApplicationQueriesSchemes	Array	(7 items)
Item 0	String	mqq
Item 1	String	mqqopensdkapi
Item 2	String	mqqopensdkapiV2
Item 3	String	mqqopensdkapiV3

将QQ的appid参数前面拼接tencent添加到info.plist文件的URL scheme, 例如appid为1106580949, 则配置如下图所示



初始化QQSDK在- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions调用

代码调用示例:

```
[JySDKManager initQQLogin:@"1106580949" universalLink:@"universalLink"];
```

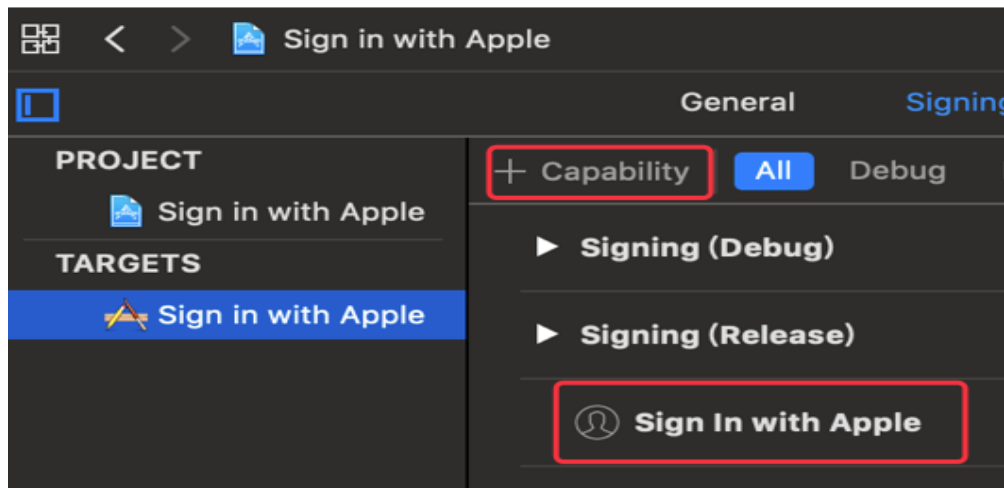
10. 使用Apple登录需要配置（选接）

1加入系统依赖库AuthenticationServices.framework

2. 登录开发者网站, 在需要添加 Sign in with Apple 功能的 Identifier 开启功能



3. Xcode 里面 Signing & Capabilities 开启 Sign in with Apple 功能



11.配置阿里云一键登录（选接）

如果要在登录界面支持显示阿里云一键登录，需在调用登录之前调用如下接口设置阿里云一键登录密钥，并且工程需配置正确的bundleid

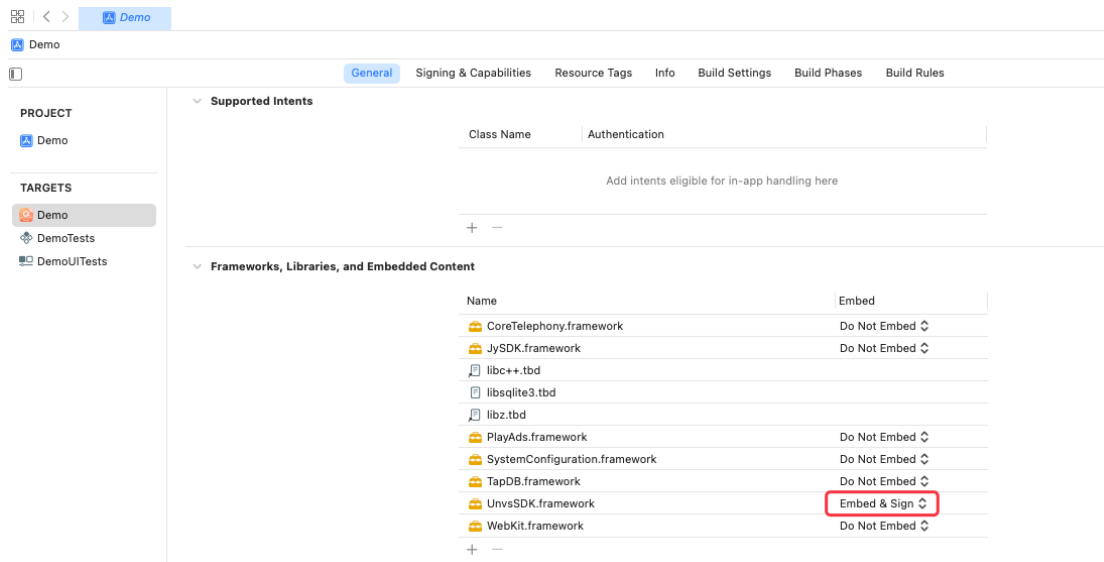
```
/**
 * @brief 设置阿里云手机号一键登录参数，需在调用SDK登录接口之前设置
 * @param authInfo 阿里云手机号一键登录密钥
 *
 */
+ (void)setPhoneFastLoginAuthInfo:(NSString *)authInfo;
```

12.配置UCloud一键登录（选接）

如果要在登录界面支持显示UCloud一键登录，需在调用登录之前调用如下接口设置UCloud一键登录APPID，并且工程需配置正确的bundleid

```
/**
 * @brief 设置UCloud手机号一键登录参数，需在调用SDK登录接口之前设置
 * @param appid UCloud手机号一键登录Appid (UCloud号码认证平台申请)
 *
 */
+ (void)setUCloudLoginWithAppId:(NSString *)appid;
```

注：UCloud登录对应的库文件UnvsSDK.framework为动态库，xcode工程里需将该文件设置为Embed & Sign。如下图所示：



```
/**
 * @brief 设置UCloud登录页屏幕方向，默认竖屏
 */
+ (void)setUCloudLoginUIInterfaceOrientation:
(UIInterfaceOrientation)faceOrientation;
```

13.配置TapTap登录（选接）

如果要在登录界面支持显示TapTap登录，需在调用登录之前调用如下接口设置TapTap登录clientId，并且工程需配置正确的bundleid

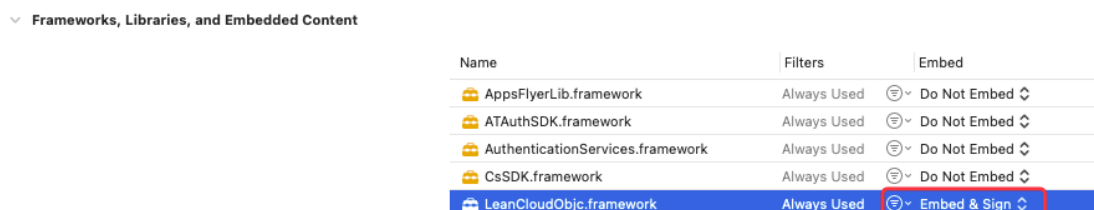
```
/**
 * @brief 设置TapTap登录参数，需在调用SDK登录接口之前设置
 * @param clientId TapTap后台clientId参数
 */
```

```
+ (void)setTapSDKWithClientID:(NSString *)clientId;
```

代码示例：

```
[JySDKManager setTapSDKWithClientID:@"s4amo8nzwsuh0itvlo"];
```

注：TapTap登录对应的库文件LeanCloudObjc.framework为动态库，xcode工程里需将该文件设置为Embed & Sign。如下图所示：



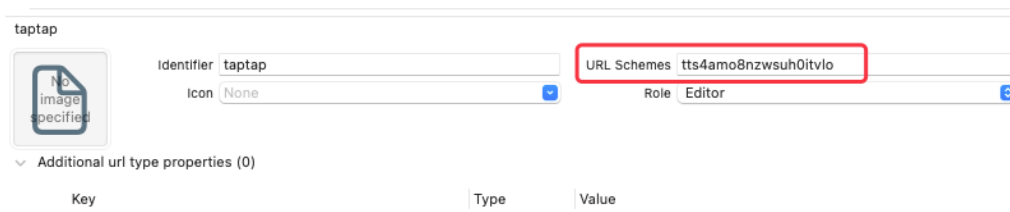
AppDelegate.m文件中必须接入以下方法

```
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
options:(NSDictionary *)options;
+ (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)source annotation:(id)annotation;
```

info.plist添加TapTap白名单 –info.plist没有LSApplicationQueriesSchemes就添加一个

▼ LSApplicationQueriesSchemes	↕	Array	(2 items)
Item 0		String	tapiosdk
Item 1		String	tapsdk

将TapTap的clientID参数前面拼接tt添加到info.plist文件的URL scheme, 例如clientID为s4amo8nzwsuh0itvlo, 则配置如下图所示



The screenshot shows the Xcode interface for configuring a URL scheme. The 'taptap' scheme is selected. The 'URL Schemes' field is highlighted with a red box and contains the value 'tts4amo8nzwsuh0itvlo'. The 'Role' is set to 'Editor'.

14. 设置角色信息（必接）

```
/**
 * 设置区服角色信息，上传角色信息（必接）
 */
+ (void)updateRoleInfo:(GameRole *)roleInfo;
```

功能：设置角色信息，在选择角色进入游戏时调用。

15. 获取用户ID（选接）

```
/**
 * 获取用户id。如未登录，返回空。
 */
+ (NSString *)userId;
```

功能：获取当前登录用户的用户ID。在用户登录成功后调用

返回值：当前登录用户的用户ID,未登录返回nil。

16. 获取用户年龄（选接）

```
/** 获取当前登录用户年龄 */
+ (NSString *)getCurrentUserAge;
```

功能：获取当前登录用户的用户年龄。在用户登录成功后调用

返回值：当前登录用户的用户年龄,未登录或者未实名返回0。

17. 获取登陆后token（选接）

```
/**
 * 获取用户校验码，用于到服务器端验证用户真实性。
 */
+ (NSString *)userToken;
```

功能：获取当前登录用户的用户token。用户登录成功时调用

返回值：当前登录用户的用户token，用以验证用户有效性,未登录返回nil。

18. 退出登录（选接）

```
/**
 * 退出登录，注销SDK账号，隐藏悬浮球
 */
+ (void)logout:(kLogoutBlock)block;
```

功能：立即将当前用户登出

设置退出登录回调方法，.h设置KAccountDelegate代理：

block用户退出回调可以传nil;

监听用户通过个人中心退出登录：

```
//设置代理，监听用户退出事件
[JySDKManager defaultManager].accountDelegate = self;
```

实现代理方法，监听用户退出和注册账号事件：

```
#pragma mark -- KAccountDelegate
- (void)userLogout:(NSDictionary *)resultDic
{
    NSLog(@"用户从个人中心手动登出。\\n%@",resultDic);
    [self login:nil];
}
- (void)userRegister:(NSString *)uid
{
    NSLog(@"注册账号: %@",uid);
}
```

用户从个人中心注销时回调，这时应该回到游戏登录界面并调用SDK的登录接口，不应该调用快捷登录接口

19. 配置是否显示游客登录（选接）

```
/** 配置是否显示游客试玩 YES显示NO隐藏默认显示*/
+ (void)configGuestLoginShow:(BOOL)show;
```

参数：show YES 不显示游客登录按钮 NO展示游客登录按钮

20. 恢复购买非消耗品接口（选接）

```
/**
```

```
恢复已购买的非消耗品或者自动订阅商品
*/
+ (void)restoreNonConsumptionProducts:(KPyResultBlock)resultBlock;
回调字典resutDic:
```

code == KorderRestoreSucc 恢复成功extInfo会附带恢复购买的productid

code == KorderRestoreFail 恢复失败 message会附带失败原因

21. 进入用户中心（选接）

```
/**
 * 调起个人中心页面
 */
+ (void)showUserCenter;
```

功能：进入用户中心页面，显示关联其他平台账号的信息和操作入口。

22. 进入客服中心（选接）

```
/**
 * 调起客服中心页面
 */
+ (void)showServiceCenter;
```

23. 悬浮窗（选接）

显示悬浮窗

```
/** 显示悬浮窗 isLeft:是否在左边 centery:中心点y值*/
+ (void)showFloatMenuBtnWithIsLeft:(BOOL)isleft andWithCenterY:
(CGFloat)centery;
```

isLeft:YES在屏幕左边NO在屏幕右边

centery:悬浮窗中心点Y坐标

隐藏悬浮窗

```
/** 隐藏悬浮窗*/
+ (void)hideFloatMenuBtn;
```

设置悬浮窗半隐藏时透明度

```
/** 设置悬浮窗隐藏时透明度alpha为0-1, 0透明1不透明*/
+ (void)configFloatAlpha:(CGFloat)alpha;
```

alpha:取值范围为0到1浮点数

24. 销毁当前账号（选接）

```
/**注销当前账号 */
+ (void)deleteCurrentAccount;
```

功能：彻底删除当前账号，不可恢复

25. 禁止自动登(选接，自动登录默认开启)

/** 设置是否自动登录，默认自动登录YES*/
+ (void)setNeedAutoLogin:(BOOL)autoLogin;
yes开启自动登录NO关闭自动登录

26. 监听实名认证回调（选接）

/** 实名认证回调须在实名认证之前设置 */
+ (void)completeRealName:(realNameResult)resultBlock;

block回调:

isComplete yes 完成实名认证 no未完成实名认证

age:用户实名认证成功返回实际年龄否则返回0

source: 调起实名认证场景 1登录调起2用户主动调起3支付调起4用户中心进入

代码调用示例:

```
[JySDKManager completeRealName:^(BOOL isComplete, NSInteger age, NSInteger source) {  
    NSLog(@"%@年龄%d", isComplete ? @"完成实名认证":@"未完成实名认证",  
    (long)age);  
}];
```

注：该接口需在用户实名认证前设置。

27. 判断用户是否实名认证（选接）

/** 用户是否已实名认证 */
+ (BOOL)isRealName;
返回值:
YES:已经实名认证;

NO:尚未实名认证;

须在登录成功之后调用。

28. 新增移除SDK内购监听（选接）

+ (void)removeListener;

tips:若游戏要自行实现内购苹果功能可调用该方法移除SDK内购监听，否则千万不要调用

29. TapDB数据统计（选接）

/**
* 初始化TapDB接口（必接），调用该接口后默认统计了初始化，登录，支付，角色信息
* appid 注册游戏时获得的APP ID
* currencyType 支付统计货币类型，参考：人民币 CNY，美元 USD；欧元 EUR，传空时默认
@"CNY"
*/
+ (void)tapDBInitWithAppID:(NSString *)appid currencyType:(NSString *)currencyType;

/**
* 自定义事件（选接），统计TapDB其他自定义事件，按需选接

```

* eventCode      事件code，需要在控制后台预先配置
* properties    事件属性，具体字段需要在控制后台预先配置
*/
+ (void)tapDBEvent:(NSString *)eventCode properties:(NSDictionary
*)properties;

```

30. AppsFlyer数据统计（选接）

```

/**
 * 初始化appsflyer（必接）
 * @param devKey AppsFlyer devKey
 * @param appID AppsFlyer appID
 * @param Sandbox 支付验证是否用沙盒模式，不传默认为NO(正式模式)，注意：出正式提审包时
一定要设置为NO
 * * @param currency 设置AF统计货币类型，可传nil或空字符串则使用默认的USD
*/
+ (void)setAppsFlyerDevKey:(NSString *)devKey appleAppID:(NSString *)appID
useReceiptValidationSandbox:(BOOL)Sandbox currencyType:(NSString *)currency;

/**
 *AppsFlyer自定义事件接口（选接）
 *
 * @param eventName 自定义事件名从AppsFlyerLib.framework的`AppsFlyerTracker.h`的
预定义常量中获取
 * @param values 参数字典（可传空），key值从`AppsFlyerTracker.h`的预定义常量中获取
调用示例：
[REDeLoginKit afTrackEvent:AFEventPurchase
withValues: @{AFEventParamRevenue : @200,
               AFEventParamCurrency : @"USD",
               AFEventParamQuantity : @2,
               AFEventParamContentId: @"092",
               AFEventParamReceiptId: @"9277"}];
*/
+ (void)afTrackEvent:(NSString *)eventName withValues:(NSDictionary *
_Nullable)values;

/** 获取AppsFlyerUID（选接）*/
+ (NSString *)getAppsFlyerUID;

```

31.热云数据统计（选接）

接入热云SDK需添加系统库：

AdSupport.framework、CoreMotion.framework、iAd.framework、libresolv.9.tbd、
libresolv.tbd、AdServices.framework(optional形式引入)

SDK内已默认集成了初始化，登录，注册，支付，游戏时长事件
相关接口如下，其中设置热云参数接口必接。

```

/**
 * 设置热云SDK参数（必接）
 * @param appKey 热云APPKEY
*/
+ (void)setRyTrackingAppKey:(NSString *)appKey;

/**

```


* 当App内的变现广告被展示时调用此方法进行事件上报（选接）

* **@param** adPlatform 填充广告的变现平台(传括号中的值)：穿山甲(csj)、优量汇(ylh)、百青藤(bqt)、快手(ks)、Sigmob(sigmob)、Mintegral(mintegral)、OneWay(oneway)、Vungle(vungle)、Facebook(facebook)、AdMob(admob)、UnityAds(unity)、IronSource(is)、AdTiming(adtiming)、游可赢(klein)

* **@param** adId 填充广告在变现平台的广告位ID。注意：若您使用聚合平台且需要通过聚合平台API获取收入时，需传聚合平台对应的值，TopOn传“adsource_id”；AdTiming传“placementId”；TradPlus传“placementId”；GroMore传“ad_unit_id”

* **@param** playSuccess 本次展示广告是否填充成功，1成功、2失败，无法确定填充是否成功时，请传1

* **@param** customParams 自定义属性key只能为string类型，名称为param1-param10，value支持字符串、数字。

```

*/
+ (void)ryTrackingAdShow:(NSString *)adPlatform adId:
(NSString *)adId isSuccess:(int)playSuccess withParams:
(nullable NSDictionary *)customParams;

/**
* 当App内的变现广告被点击时调用此方法进行事件上报（选接）
* @param adPlatform 填充广告的变现平台(传括号中的值)：穿山甲(csj)、优量汇(ylh)、百青藤(bqt)、快手(ks)、Sigmob(sigmob)、Mintegral(mintegral)、OneWay(oneway)、Vungle(vungle)、Facebook(facebook)、AdMob(admob)、UnityAds(unity)、IronSource(is)、AdTiming(adtiming)、游可赢(klein)
* @param adId 填充广告在变现平台的广告位ID。注意：若您使用聚合平台且需要通过聚合平台API获取收入时，需传聚合平台对应的值，TopOn传“adsource_id”；AdTiming传“placementId”；TradPlus传“placementId”；GroMore传“ad_unit_id”
* @param customParams 自定义属性key只能为string类型，名称为param1-param10，value支持字符串、数字。
*/
+ (void)ryTrackingAdClick:(NSString *)adPlatform adId:
(NSString *)adId withParams:(nullable NSDictionary
*)customParams;

/**

```

* 热云自定义事件：用于统计用户在应用内的任意行为，如打开某个面板、点击某个Button、参与某个活动等。（选接）

* @param eventName 自定义事件名称，必须为event_1到event_30，长度最长32位

* @param customParams 自定义属性key只能为string类型，名称为param1-param10，value支持字符串、数字。

*/

```
+ (void)ryTrackingEvent:(NSString *)eventName param:
(nullable NSDictionary *)customParams;
```

/** 获取热云DeviceId（选接）*/

```
+ (NSString *)getRyDeviceId;
```

32.微博分享（选接）

/**

设置微博参数

@param appkey 微博APPKEY

@param universalLink 微博universalLink

@param redirectURL 微博redirectURL

*/

```
+ (void)initWbWithAppKey:(NSString *)appkey
universalLink:(NSString *)universalLink redirectURL:
(NSString *)redirectURL;
```

/**

微博分享文本消息

@param text 分享的文本内容，长度小于2000个汉字

@param completion 分享结果回调 0分享成功，-1分享取消 -2分享失败

*/

```
+ (void)wbShareMessageToText:(NSString *)text
shareResult:(kOtherResultBlock)completion;
```

/**

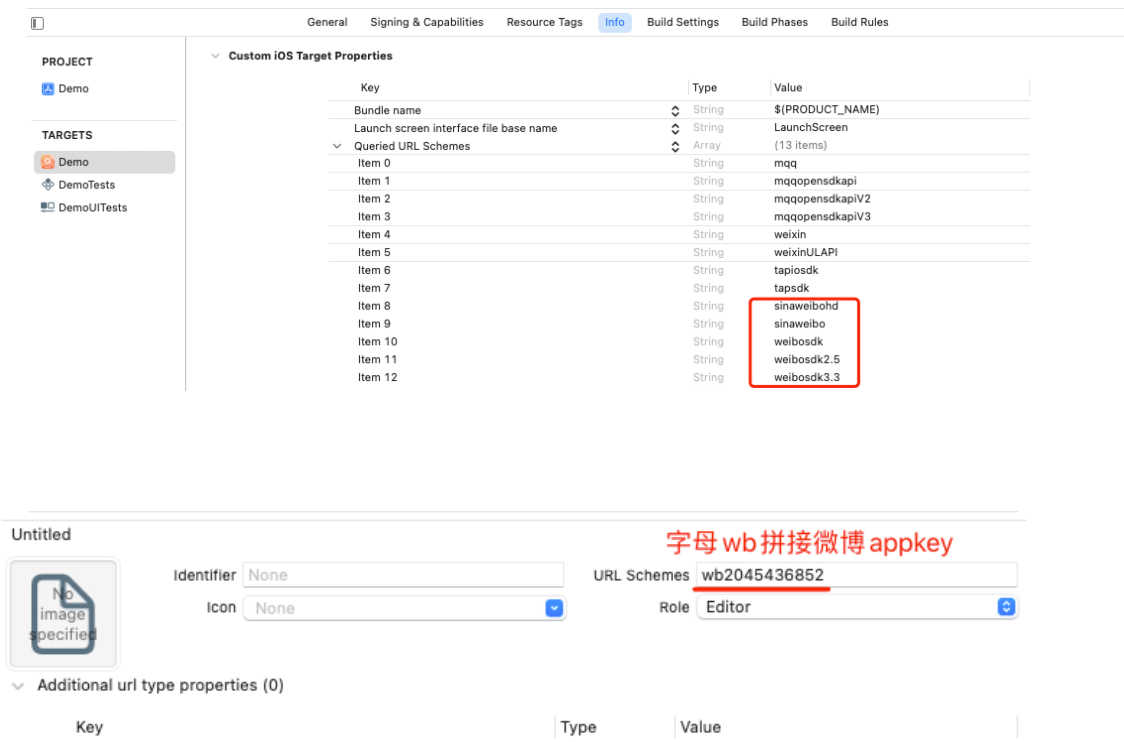
微博分享图片

@param image 要分享的图片对象，图片格式png，不能为空且大小不能超过10M

@param completion 分享结果回调 0分享成功，-1分享取消 -2分享失败

*/

```
+ (void)wbShareMessageToImage:(UIImage *)image
shareResult:(kOtherResultBlock)completion;
```



33.腾讯云一键登录（选接）

如果要在登录界面支持显示腾讯云一键登录，需在调用登录之前调用如下接口设置腾讯云一键登录appId，并且工程需配置正确的bundleid

```
/**
 * @brief 设置腾讯云手机号一键登录参数，需在调用SDK登录接口之前
设置
 * @param appid 腾讯云手机号一键登录Appid（腾讯云平台申请）
 */
+ (void)setTengXunLoginWithAppId:(NSString *
Nonnull)appid;
```

五、自定义UI配置（选接）

描述：**sdk**支持自定义背景图以及自定义文本颜色

登录界面顶部默认为文字标题（登录），可自行更改为**logo**图片

logo位置顶部居中。名称设为**logo_head @2x.png**, **logo_head @3x.png**放置到**jywlRes.bundle**文件夹内即可。

logo_head @2x.png 尺寸为：320*72

logo_head @3x.png 尺寸为：480*108

Bundle里面图片都可以替换，找到需要替换的图片，新建一个同名同尺寸的图片替换即可。其中**sdk**背景图片名称前缀为**log_back**

自定义颜色设置

颜色设置统一采用**16**进制。

在**bundle**中找到**ColorStyle.plist**文件，修改对应位置颜色设置。建议根据背景图片来配置。