### iOS SDK接入文档2018-3-13

#### 谁来阅读此文档

接入厂商的产品，技术人员，平台技术人员。

#### 保存机制 ——— 首先根据keychian获取(首次安装的肯定没有) ,keychian中无账号数据，则 根据key遍历本地存储记录 遍历存储记录和服务器返回的, 做排同处理(相同账号只保留有密码的那一个)。 在每次登录或注册成功都会将所有用户账号密码数据写入keychian中 !

#### 注意！！！！

本文档新加keychian sharing 功能 sdk配置文件中appKeyChian字段为servername 一般以包名开头后加自定义字符即可

如果是上线到appstore，SDK有使用到**IDFA**，提交审核的时候请注意选上。  
勾选详情参照http://www.jianshu.com/p/3ef714214405

#### 本SDK不需要任何第三方支付白名单和第三方支付库依赖！都不需要添加！

#### 为规避苹果第三方检测，第三方支付特从浏览器开启返回应用，不要按手机Home键返回应用，SDK在浏览器页面提供了返回按钮！！！

已经集成过老版本的注意

#### 只需要在老版本的基础上修改相关配置

1 替换Framework文件 按照要求填写sdk中plist文件中的 apple\_id ,apple\_id为苹果注册APP id以前的参数废弃不用  
2 删掉有关第三方支付的白名单和相关第三方支付库(删除info.plis中LSApplicationQueriesSchemes的相关第三方支付白名单)  
3 删掉ThirdLibs文件夹

4需在info.plist中添加相册访问权限

### 不懂就把两个都加上把(特别是Xcode 9 开发的同学记得仔细看下)

加上NSPhotoLibraryAddUsageDescription和NSPhotoLibraryUsageDescription(后面一定要加上应用获取相册权限的描述) 原因如下：

相册权限选择介绍<https://stackoverflow.com/questions/46344159/whats-difference-between-nsphotolibraryaddusagedescription-and-nsphotolibraryus>  
NSPhotoLibraryUsageDescriptio 支持iOS8+  
NSPhotoLibraryAddUsageDescription 只支持iOS11+  
需要加上两个(NSPhotoLibraryUsageDescriptio,  
NSPhotoLibraryAddUsageDescription)描述避免崩溃  
[如：App需要您的同意,才能访问相册以保存您的注册账号信息])

#### 切换支付的sdk才有的

5 外部跳转项目协议CP会提供一个跳转协议的参数，直接复制粘贴(位置在info.plist中的CFBundleURLTypes中的CFBundleURLSchemes)

以上为接入过sdk人员参考

#### 一.SDK说明与集成

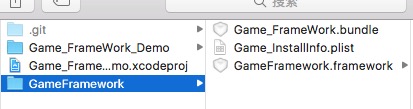
本SDK支持iOS7及以上系统版本，兼容横竖版。  
支持armv7 i386 x86\_64 arm64

##### 1.1.获取参数

从平台处，获取以下参数，用于对接，配置到sdk的Game\_InstallInfo.plist文件中

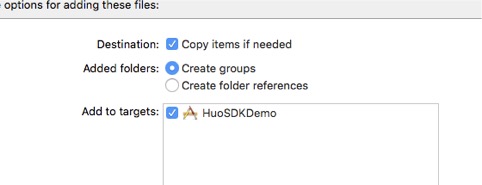
| **参数名** | **说明** |
| --- | --- |
| apple\_id | 苹果官网注册的apple\_id |
| isLandscape | 是否横屏，1为横屏，0为竖屏 |

##### 1.2.SDK的组成，如下图所示。



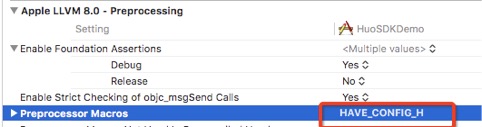
Game\_FrameWork.bundle:sdk所使用的资源  
GameFramework.framework:sdk的静态库和头文件  
Game\_InstallInfo.plist:SDK的配置文件

##### 1.3.SDK集成和配置说明

1.将SDK拖入工程主目录下  


2.TARGETS->Build Settings->other linker :-ObjC 和-lsqlite3.0  

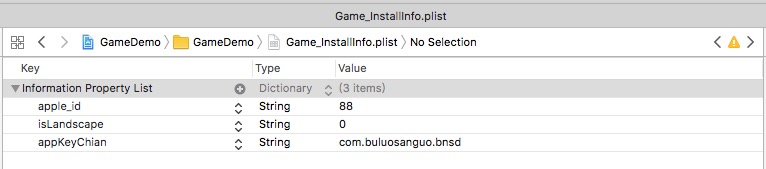

3.TARGETS->Build Settings->Enable Bitcode设置为No

4.在Preprocessor Macros 设置HAVE\_CONFIG\_H(在build Settings直接搜索即可)  


5.TARGETS->Build phases->Link Binary With Libraries链接以下系统库  
(1)libz.1.2.5  
如果sdk使用了第三方登录，则还需要链接以下库  
(2)ImageIO  
(3)libicucore  
(4)libiconv  
(5)libsqlite3  
(6)libstdc++  
删掉其他所有有关第三方支付的库

6.工程info.plist添加以下配置第三方信息需要自行申请,实例只供参考

1. <**key**>NSAppTransportSecurity</**key**>
2. <**dict**>
3. <**key**>NSAllowsArbitraryLoads</**key**>
4. <**true**/>
5. </**dict**>
6. 如果是有第三方登录，还需增加微信，qq，微博的配置，参数请从运营方获取
7. ```xml
8. <**key**>CFBundleURLTypes</**key**>
9. <**array**>
10. <**dict**>
11. <**key**>CFBundleURLSchemes</**key**>
12. <**array**>
13. <**string**>h1tsdk.com11</**string**>（跳转协议位置）
14. </**array**>
15. </**dict**>
16. <**dict**>
17. <**key**>CFBundleTypeRole</**key**>
18. <**string**>Editor</**string**>
19. <**key**>CFBundleURLName</**key**>
20. <**string**>weixin</**string**>
21. <**key**>CFBundleURLSchemes</**key**>
22. <**array**>
23. <**string**>wx9efca7e13875a222</**string**>
24. </**array**>
25. </**dict**>
26. <**dict**>
27. <**key**>CFBundleTypeRole</**key**>
28. <**string**>Editor</**string**>
29. <**key**>CFBundleURLSchemes</**key**>
30. <**array**>
31. <**string**>tencent101379012</**string**>
32. </**array**>
33. </**dict**>
34. <**dict**>
35. <**key**>CFBundleTypeRole</**key**>
36. <**string**>Editor</**string**>
37. <**key**>CFBundleURLSchemes</**key**>
38. <**array**>
39. <**string**>wb1220169449</**string**>
40. </**array**>
41. </**dict**>
42. </**array**>

7.SDK配置文件Game\_InstallInfo.plist说明有且只有下面两个参数，这些参数不清楚可以询问工作人员  


\*\*注意参数类型一定要都是String类型  
apple\_id:苹果管网注册的应用id

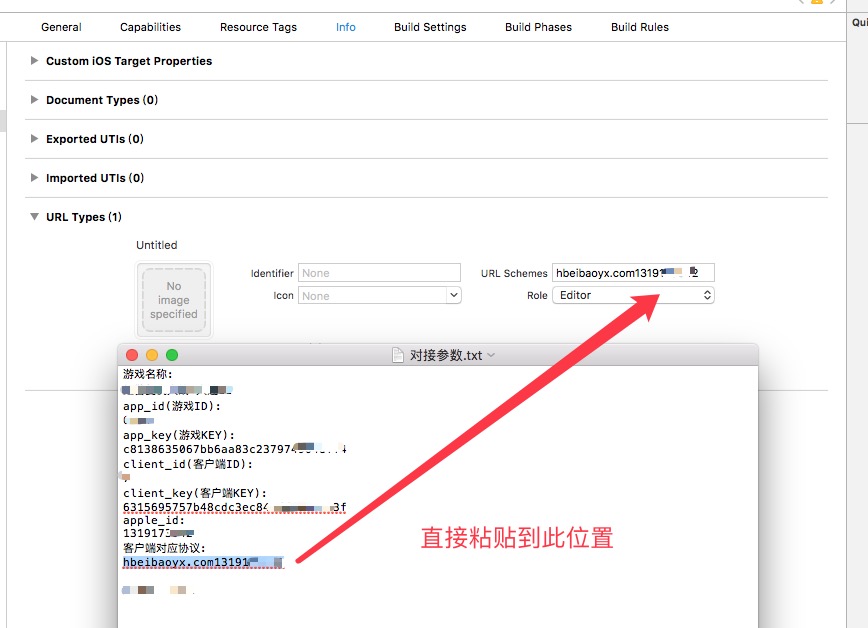
isLandscape:是否横屏，1为横屏，0为竖屏

appKeyChian :keychian 标示一般以包名开头填写后加自定义字符即可

URLSchemes: cp会提供一个叫做跳转协议的参数一般直接复制到下图中位置

注：跳转协议CP会提供的，直接粘贴到下图位置

#### 无切换支付的不用添加协议

根据Game\_InstallInfo.plist的参数，在应用的info.plist配置一个url schemes参数，用于应用的跳转  


除上述说明的参数其他无（不修改)

#### 二.SDK接口说明

使用SDK之前，需要#import#<GameFramework/GameFramework.h>

appdelegate方法，需要在AppDelegate.m加入SDK的这个方法

如若不添加此处函数实现，那么从浏览器支付之后回到应用会收不到支付结果

1. - (**BOOL**)application:(**UIApplication** \*)application openURL:(**NSURL** \*)url sourceApplication:(**NSString** \*)sourceApplication annotation:(**id**)annotation
2. {
3. [Game\_Api Game\_application:application openURL:url **source**Application:**source**Application annotation:annotation];
4. **return** YES;
5. }
6. //iOS10以上使用
7. - (**BOOL**)application:(**UIApplication** \*)app openURL:(**NSURL** \*)url options:(**NSDictionary**<UIApplicationOpenURLOptionsKey,**id**> \*)options
8. {
9. [Game\_Api Game\_application:appn openURL:url sourceApplication:**nil** annotation:**nil**];
10. **return** YES;
11. }
12. - (**BOOL**)application:(**UIApplication** \*)application handleOpenURL:(**NSURL** \*)url
13. {
14. [Game\_Api Game\_application:application openURL:url sourceApplication:**nil** annotation:**nil**];
15. **return** YES;
16. }

##### 2.1.登录

只需调用以下代码即可进入SDK的登入流程，当登入成功，SDK会自动关闭SDK页面，并且回调用户信息给游戏。

1. [Game\_Api Game\_showLoginWithCallBack:^(NSDictionary \*responseDic) {
2. //**block**激活即是SDK登录成功，这个时候游戏通过**block**获取用户信息
3. **NSString** \*sdk\_userid = responseDic[@"userid"];//sdk的用户id
4. **NSString** \*sdk\_sessionid = responseDic[@"token"];//sdk的sessionid，用于验证登录是否成功
5. **NSLog**(@"userid = %@", sdk\_userid);
6. **NSLog**(@"token = %@", sdk\_sessionid);
7. useridLabel.**text** = [NSString stringWithFormat:@"userid=%@", sdk\_userid];
8. sessionLabel.**text** = [NSString stringWithFormat:@"token=%@", sdk\_sessionid];
9. }];

//SDK登录接口，只有在登录成功的时候才会激活回调，登录失败则由sdk处理  
+ (void)Game\_showLoginWithCallBack:(GameMainThreadCallBack)receiverBlock;

##### 2.2.登出

//登出接口，当用户在游戏菜单登出成功的时候请调用该方法  
+ (void)Game\_Logout;

//登出回调接口，有两种情况会激活该block，1.用户在游戏内进行登出，2.用户在SDK的菜单进行登出成功，假如是从SDK发起登出的，请在block激活的时候对游戏进行登出操作  
+ (void)addLogoutCallBack:(HuoSDKMainThreadCallBack)receiverBlock;  
登出成功之后应该调出SDK的登录窗口

###### 2.3. 支付接口

//支付接口  
+(void)Game\_sendOrderInfo:(NSDictionary\*)info failedBlock:(void(^)())failedBlock;

1. **NSDictionary** \*orderInfo = @{
2. element\_cpOrderid:@"testorderid",
3. element\_serverID:@"testserver",
4. element\_serverName:@"testserverName",
5. element\_productID:@“testproductid",
6. element\_productName:@"testgoods",
7. element\_productdesc:@"goodsdes",
8. element\_ext: @"testattach",
9. element\_productPrice:@“1.0”,
10. element\_roleID:@"testrole",
11. element\_roleName:@"testroleName",
12. element\_currencyName : @"123",
13. };
14. [Game\_Api Game\_sendOrderInfo:orderInfo failedBlock:^{
15. **NSLog**(@"payfailed");//支付失败
16. }];

| **参数名** | **说明** |
| --- | --- |
| element\_cpOrderid | 游戏使用的订单号 |
| element\_serverID | 角色所在的serverid |
| element\_serverName | 服务器名 |
| element\_productID | 商品id |
| element\_ext | 扩展参数,可选 |
| element\_productPrice | 商品金额(单位：元) 建议传入整数,可以保留两位小数 |
| element\_roleID | 角色id |
| element\_roleName | 角色名字 |
| element\_productName | 商品名字 |
| element\_productdesc | 商品描述，可选 |
| element\_currencyName | 货币名，如金币,元宝,钻石等。 |

调用示例

监听支付成功的回调，需要在调用支付之前调用，支付成功之后会返回平台订单号和研发的订单号。

1. [Game\_Api Game\_sendInfoSuccessedCallBlock:^(NSDictionary \*responseDic) {
2. **NSString** \*orderid = responseDic[@"orderid"];//冰狐订单号
3. **NSString** \*cporderid = responseDic[@"cpi"]; //cp订单号
4. }];

###### 2.5. 设置角色信息接口

该接口在能获取以下信息的时候再进行设置即可，一般是登录成功之后。

参数说明

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| element\_serverID | String | 角色所在的serverid |
| element\_serverName | String | 服务器名 |
| element\_partyName | String | 公会名字，可选 |
| element\_roleID | String | 角色id |
| element\_roleName | String | 角色名字 |
| element\_roleLevel | String | 角色等级，如果没有，请填@”0” |
| element\_roleVip | String | VIP等级，如果没有，请填@”0” |
| element\_roleBalence | String | 角色游戏币余额，请填@”0.0” |
| element\_dataType | String | 数据类型，1为进入游戏，2为创建角色，3为角色升级，4为退出 |
| element\_rolelevelCtime | String | 创建角色的时间 时间戳(无则填大等于1的数字) |
| element\_rolelevelMtime | String | 角色等级变化时间 时间戳(无则填大等于1的数字) |

//设置角色信息  
+(void)Game\_setRoleInfo:(NSDictionary\*)roleInfo;

调用示例

1. [Game\_Api Game\_**set**RoleInfo:@{element\_dataType : @"1",
2. element\_serverID : @"s1",
3. element\_serverName : @"testserver",
4. element\_roleID : @"123",
5. element\_roleName : @"iOS测试账号",
6. element\_roleLevel : @1,
7. element\_roleVip : @3,
8. element\_roleBalence : @12.3,
9. element\_partyName : @"testparty",
10. element\_rolelevelCtime : @"1479196021",
11. element\_rolelevelMtime : @"1479196736",
12. }];

###### 2.6. 设置是否一键登录

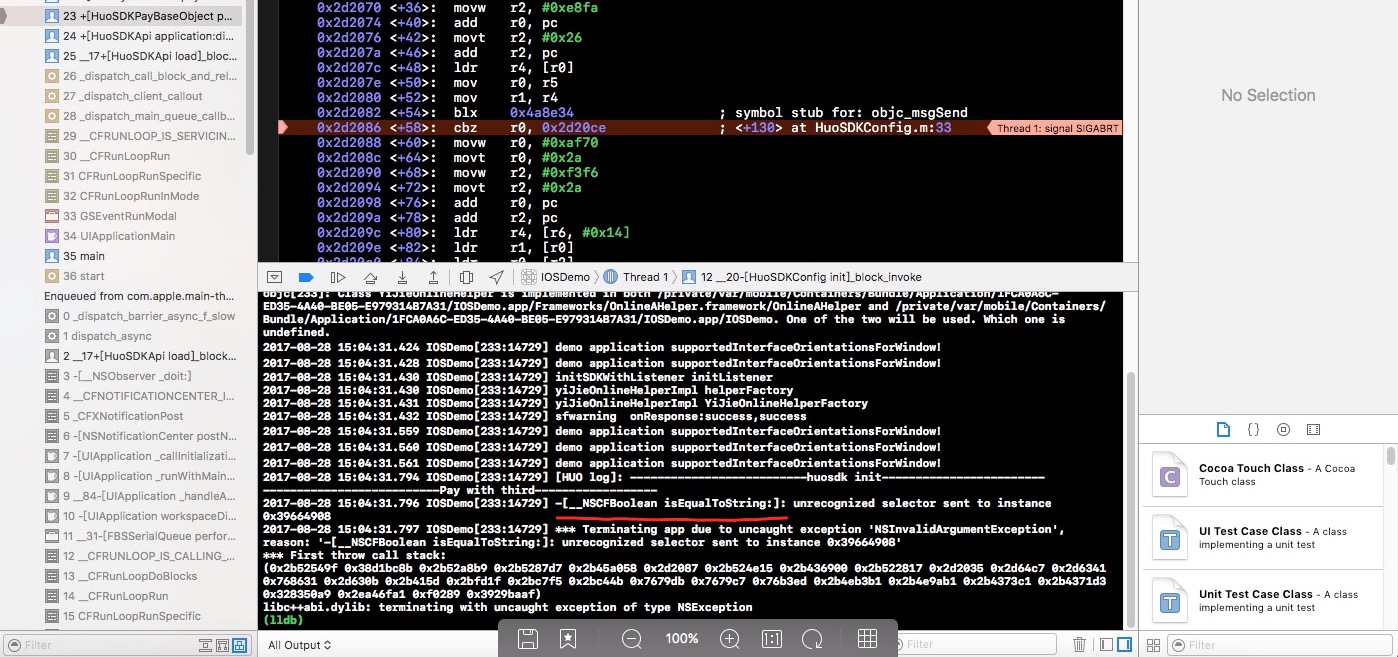
说明：默认为不使用一键登录，如果需要此功能，如下调用  
[Game\_Api Game\_setIsAutoLogin:YES];  
[Game\_Api Game\_showLoginWithCallBack:^(NSDictionary \*responseDic) {  
NSLog(@”登录成功”);  
//block激活即是SDK登录成功，这个时候游戏通过block获取用户信息  
}

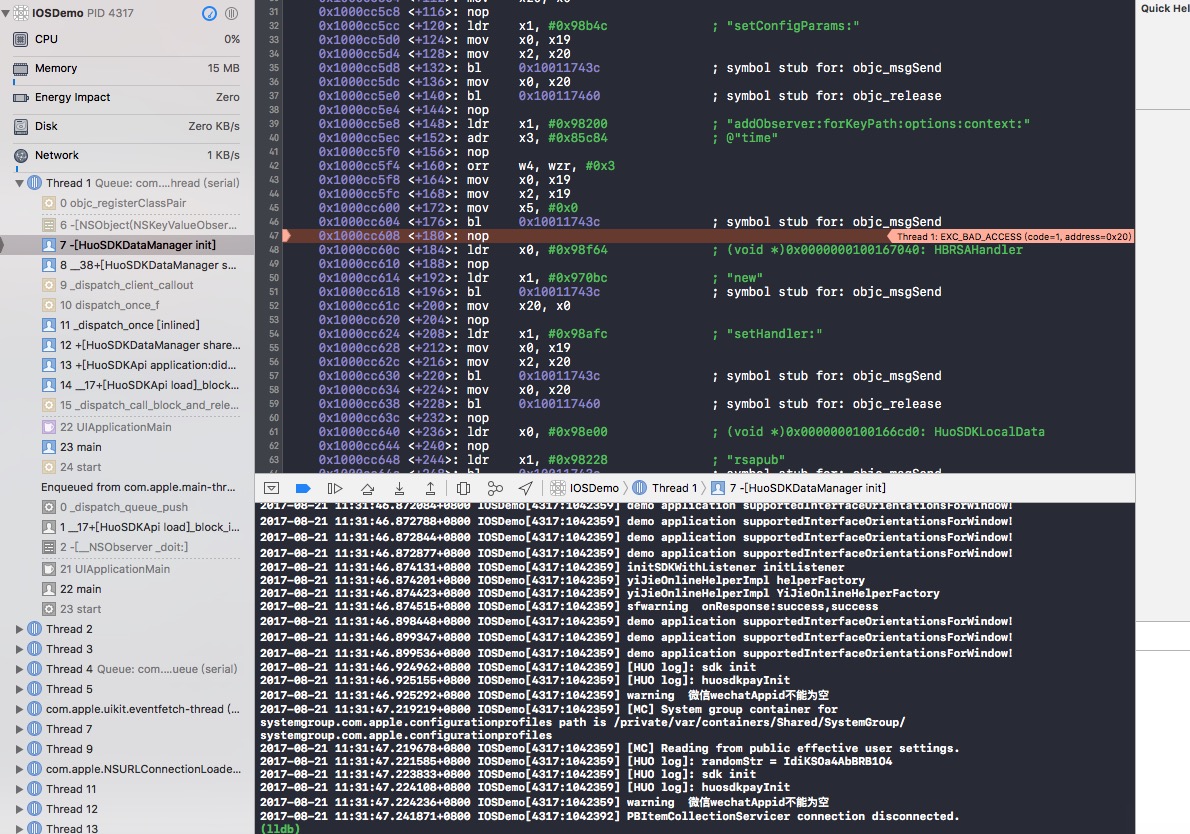
如出现加载登录界面出现CPU过大

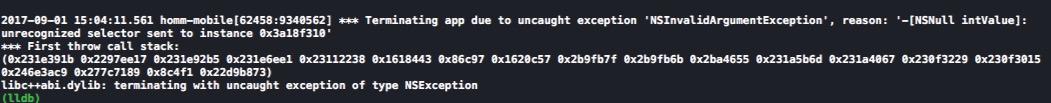
###### 需在调用登录方法获取登录验证结果处发送通知

[[NSNotificationCenter defaultCenter] postNotificationName:@”endTransFormAnimationWithNotification” object:nil];

\*\*\*注：常见错误

1错误原因： 未按照文档填入SDK配置文件参数，类型填入错误！都要填入String类型  
崩溃日志：

2错误原因：重复导入SDK和配置文件(易接一般犯错)本身封装了冰狐sdk，然后又在项目中导入冰狐sdk,需要删除其中一个的关联(找不到两个就删掉能看到的那个)  
崩溃日志：

3错误原因：在登录成功的回调中设置了显示浮点，未设置对应的横竖屏 需要在sdk配置文件中设置对应的横竖屏  
崩溃日志:

4错误原因：填入应用的apple\_id没有在管理后台配置，需要配置获取验证参数给后台。客户端只要填入了对的apple\_id是不会有问题了，验证问题跟客户端无关！！！  
错误提示：客户端返回app\_id错误