

HJSDK 接入文档 【奇葩版本，新增奇葩商城功能】

SDK 介绍，此文档只适用于出奇葩渠道包

> SDK 内涵两个文件，直接拷贝所有的文件到项目中（包含Plist 配置文件，Framework 文件夹）

项目配置

> info.plist 文件配置，直接拷贝以下key/value 值到项目中info.plist 文件内

“

```
<key>CFBundleURLTypes</key>
<array>
<dict>
<key>CFBundleTypeRole</key>
<string>None</string>
<key>CFBundleURLName</key>
<string>id1</string>
<key>CFBundleURLSchemes</key>
<array>
<string>As3713</string>
</array>
</dict>
<dict>
<key>CFBundleTypeRole</key>
<string>Editor</string>
<key>CFBundleURLName</key>
<string>id2</string>
<key>CFBundleURLSchemes</key>
```

```
<array>
<string>Demo.GPGameSDK-SDKDemo.guopan</string>
</array>
</dict>
</array>
<key>CFBundleVersion</key>
<string>1</string>
<key>LSApplicationQueriesSchemes</key>
<array>
<string>mqqapiwallet</string>
<string>i4Tool4008227229</string>
<string>mqqwpa</string>
<string>alipays</string>
<string>alipayshare</string>
<string>alipayqr</string>
<string>XXAppstore</string>
<string>xxassistantsdkV2</string>
<string>cydia</string>
<string>xxassistant</string>
<string>xxassistantsdk</string>
<string>alipay</string>
<string>weixin</string>
<string>wechat</string>
<string>weixinULAPI</string>
<string>mqqwallet</string>
<string>rnotes</string>
</array>
<key>LSRequiresiPhoneOS</key>
<true/>
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
<key>NSCameraUsageDescription</key>
<string>使用相机</string>
<key>NSLocationAlwaysUsageDescription</key>
<string>需要使用地理位置信息</string>
<key>NSLocationWhenInUseUsageDescription</key>
<string>需要使用地理位置信息</string>
<key>NSMicrophoneUsageDescription</key>
<string>需要使用麦克风</string>
<key>NSPhotoLibraryAddUsageDescription</key>
<string>需要使用相册</string>
```

```
<key>NSPhotoLibraryUsageDescription</key>  
<string>需要使用相册</string>
```

> SDK 动态库设置，不设置启动会崩溃（切记）

> Target - Build Phases - Embed Frameworks 选项中引用

HJAggregationSDK.framework 一个动态库（或是在Target - General - Embedded Binaries 中引入上面一个动态库）

> 添加奇葩聚合 SDK 和 quick 聚合 SDK（libSMPCQuickChannel.a 静态库, QPJHLightQcSDK.framework 静态库, SMPCQuickSDK.framework 静态库）

> Target - Build Phases - Link Binary With Libraries 选项中引用

libSMPCQuickChannel.a 静态库, QPJHLightQcSDK.framework 静态库, SMPCQuickSDK.framework 静态库静态库

> 还需要引入 QPJHLight.bundle 资源文件和 ChannelBridge.h 类文件到项目中

> 设置 Bitcode 为 No

> Target - Build Settings - Enable Bitcode 选项设置为 NO

> 在 Build Settings 中将, 将 Allow Non-modular Includes In Framework Modules 设置为 YES。

> Architectures 设置中移除 armv7 架构, 不需要支持 iPhone5 以下设备。

> Other Linker Flags 中添加-ObjC

SDK 参数处理（提供就填写，不提供无需修改，默认 plist 文件数据即可）

GamePlbulic.plist 文件配置

> 只需要更改 Plist 文件夹中 GamePlbulic.plist 中对应的数据

> App_id: 聚合的游戏 ID

> App_ChannelId: 未提供不修改，提供了更改成对应的数据

GameParametersList.plist 文件配置，只需要加入此文件，无需修改文件内参数

QPJHPLIST.plist 文件配置

> QPJHPLIST.plist 文件其中三个参数都会提供，替换对应参数即可

代码处理（奇葩商城接口需要调用奇葩对外接口，再最后有文档，根据运营需求是否需要接入）

遵循代理（根据代理接受回调数据的状态）

> 需要在对应的类中，实现代理方法，下面会说明

> 需要遵守代理<JHAggregationInitDelegate>

'''

//遵循代理

[HJAggregationSDK sharedJHAggregation].delegate = self;

'''

初始化

'''

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary
*)launchOptions {
```

//初始化聚合接口

```
[[HJAggregationSDK sharedJHAggregation]
initAggregationapplication:application
didFinishLaunchingWithOptions:launchOptions];
```

```
return YES;
```

```
}
```

SDK 登录

```
[[HJAggregationSDK sharedJHAggregation] loginGame];
```

角色上报

> 参数必传参数必传，若没有统一传@"1"。

```
NSMutableDictionary *userDict = @{
    @"key_dataType" : @"0", //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，5 为充值
    @"key_serverId" : @"0", //角色所在的 serverid
    @"key_serverName" : @"剑域 1 服 00", //服务器名
    @"key_roleId" : @"10000010", //角色 id
    @"key_roleName" : @"空若冰 qq", //角色名字
    @"key_roleLevel" : @"1", //角色等级
    @"key_roleVip" : @"1", //角色 VIP 等级
    @"key_roleBalance" : @"0", //角色游戏币余额
    @"key_partyName" : @"testparty", //公会名字，可选
    @"key_rolelevelCtime" : @"0", //创建角色的时间 时间戳（秒级别）
    @"key_rolelevelMtime" : @"0", //角色等级变化时间 时间戳
    @"key_currencyName": @"元宝" //货币名
};
[[HJAggregationSDK
sharedJHAggregation]userInfoGame:userDict];
```

支付

> 参数必传，若没有统一传@"1"。

```
NSDate * date = [NSDate date];
NSDateFormatter * dateFormatter = [[NSDateFormatter alloc]
init];
[dateFormatter setDateFormat:@"YYYY/MM/ddHH:mm:ss"];
NSString * dateStr = [dateFormatter stringFromDate:date];

NSDictionary *orderInfo = @{
@"key_cpOrderId": dateStr,          //游戏使用的订单号，必须唯一
@"key_serverId": @"1", //角色所在的 serverId
@"key_serverName": @"剑域 1 服", //服务器名称
@"key_productId": @"2", //商品 ID(内购代码)
@"key_productName": @"6", //商品名字
@"key_productdesc": @"6", //商品描述(可选)
@"key_ext": @"1_80127", //扩展参数，CP 订单，默认传@"1"。
@"key_productPrice": @"1.00", //商品金融（单位元，要精确到小数点
后两位数，比如 1.00）
@"key_roleId": @"10000010", //角色 ID
@"key_roleName": @"空若冰 ads", //角色名字
@"key_currencyName" : @"元宝", //货币名
@"key_callbackurl": @" " //支付回调地址
};
[[HJAggregationSDK
sharedJHAggregation]orderInfoGame:orderInfo];
```

登出（退出登录）

```
[[HJAggregationSDK sharedJHAggregation]loginOutGame];
```

JHAggregationInitDelegate 代理回调实现

> 支付回调，一切以服务端结果为准

```
/**
初始化成功的回调
@param initDict 初始化的返回数据
*/
- (void)inInitDict:(NSDictionary *)initDict{
NSLog(@"初始化成功%@",initDict);
}

/**
初始化失败的回调
@param initDict 初始化失败的返回数据
*/
- (void)InInitDictFailed:(NSDictionary *)initDict{
NSLog(@"初始化失败%@",initDict);
}

/**
登录成功的回调
@param userDict 登录的返回数据
*/
- (void) loginSuccendUserDict:(NSDictionary *)userDict{

NSLog(@"用户 UserID:%@",userDict[@"UserId"]);//用户 ID

NSLog(@"用户 Toekn:%@",userDict[@"Token"]);//用户 Token

NSLog(@"验证地址 Url:%@",userDict[@"Url"]);//验证登录的地址

}

/**
```



```
登录失败的回调
@param userDict 登录失败的返回数据
*/
- (void) loginFailedUserDict:(NSDictionary *)userDict{
NSLog(@" 登录失败的回调!");
}

/**
登出的回调
@param dict 登出的返回数据
*/
- (void) loginOutSuceend:(NSDictionary *)dict{
NSLog(@"登出回调成功!");
}

/**
登出失败回调
@param dict 登出失败的返回数据
*/
- (void) loginOutFailed:(NSDictionary *)dict{
NSLog(@"登出失败回调!");
}

/**
上报角色成功的回调
@param infoDict 上报角色的返回数据
*/
- (void) uploadSuccendInfoDict:(NSDictionary *)infoDict{
NSLog(@"上传角色回调成功!");
}

/**
上报角色失败的回调
@param infoDict 上报角色失败的返回数据
*/
- (void) uploadFailedInfoDict:(NSDictionary *)infoDict{
NSLog(@"上报角色失败的回调!");
}

'''
```

奇葩商城接口，需要引用头文件#import

<QPJHLightQcSDK/QPJHLightQcSDK.h>

> 需要遵守奇葩 SDK 的代理<JHLCoreEntityDelegate>

> 在对应需要调用奇葩商城的地方调用此方法

> 此功能需要与运营核对使用

#pragma mark - 奇葩商城，非必须，是否接入徐找运营确认

```
- (void)qipaShangchengButton:(UIButton *)sender {  
    //设置奇葩代理  
    [JHLCoreEntity sharedInstance].delegate = self;  
    //打开奇葩商城  
    [[JHLCoreEntity sharedInstance] qpjh_openGmShop];  
}
```

> 下面是奇葩商城方法的代理回调代码

#pragma mark - JHLCoreEntityDelegate

//奇葩活动商城充值回调

```
- (void)rechargeCallBack:(NSString *)result {  
    NSLog(@"活动商城充值回调:%@", result);  
    #warning - 请在该方法中打开游戏充值档位页面  
}
```
