# HJSDK接入文档

## SDK介绍

> SDK内涵三个文件，直接拷贝所有的文件到项目中（包含Plist配置文件，Framework动态库，Bundle资源文件）

## 项目配置

> 横竖屏配置，由于爱思渠道需要配置Portrait

>Target - Deployment Info - Device Orientationq中横竖屏全部勾选（Portrait,Upside Down,Landscaoe Left,Landscaoe Right）

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

![截图](https://github.com/zhongaiyemaozi/HJSDK/blob/master/Resources/Sources/InfoPlist.png)

```

<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>openedvv</string>

<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>

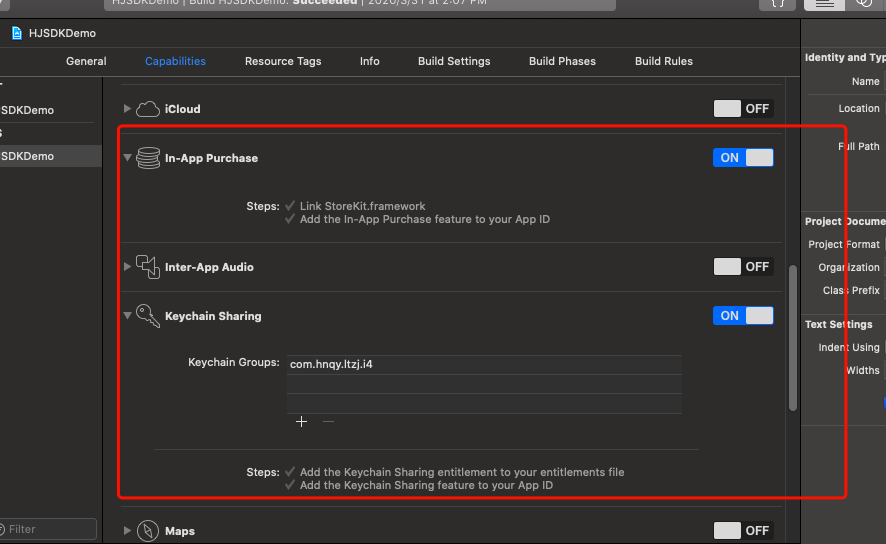
<string>需要使用相册</string>

```

>Capabilities设置

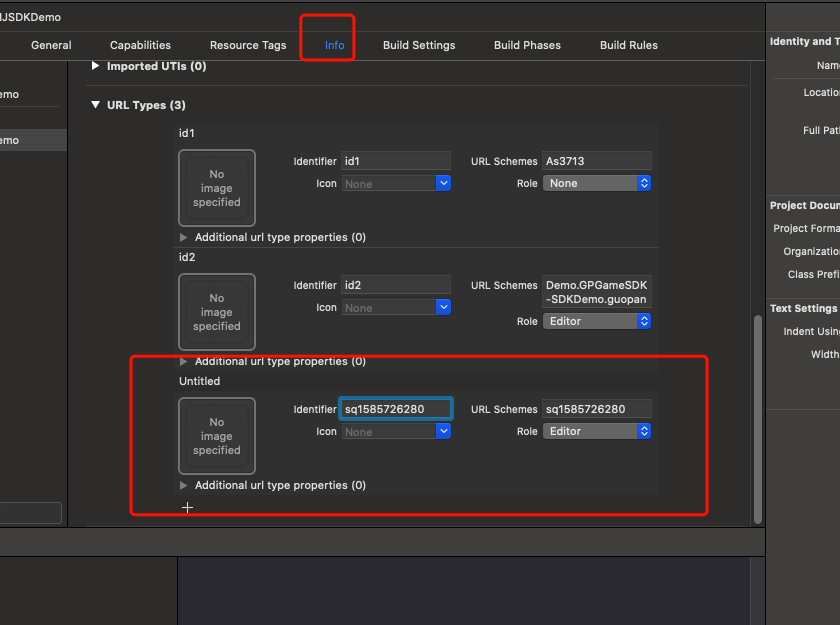
>使用苹果内购支付时，需在 Xcode 工程的 Targets ->Capabilities -> 打开 In-App Purchase

>由于 SDK 内部加密协议涉及 KeyChain，如使用 Xcode8 及以上环境开发，在模拟器运行调试时请开启 Application Targets 的 Capabilities->KeyChain Sharing 选项



>添加VV平台跳转

>添加对游戏的跳转支持,需在 Xcode 工程的 Targets ->Info ->URL Types（会提供对应的iOS识别码）



> 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DynamicLibrary.framework，SMSDK.framework，HJAggregationSDK.framework三个动态库（或是在Target - General - Embedded Binaries中引入上面三个动态库）

![Embed](https://github.com/zhongaiyemaozi/HJSDK/blob/master/Resources/Sources/EmbedFrameworks.png)

> 设置Bitcode为No

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

>Target - Other Linek - 设置ObjC

![Bitcode](https://github.com/zhongaiyemaozi/HJSDK/blob/master/Resources/Sources/Bitcode.png)

## SDK参数处理

### 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\_Name：App名称

>App\_id：游戏ID

>App\_Channelld：未提供不更改（默认131），提供了更改成对应的数据

![GamePlbulic](https://github.com/zhongaiyemaozi/HJSDK/blob/master/Resources/Sources/GamePlbulic.png)

### GameParametersList.plist文件配置

![GamePlbulic](https://github.com/zhongaiyemaozi/HJSDK/blob/master/Resources/Sources/GameChannePlist.png)

## 代码处理（只需要对接HJAggregationSDK.framework的对外接口）

### 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"。

```

NSDictionary \*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\_roleBalence" : @"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

> 支付回调是预留接口，支付结果SDK此版本不回调，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!");

}

```