

# OPPO 游戏中心\_SDK 集成说明书 ( 单机 )

- OPPO -

文档编号	300001	文档版本号	V1.0
归属部门	游戏中心	文档机密	中
产品名	游戏 SDK	产品版本	v2.0
编写人	卜艳丽	编写日期	2015-10-22

## 修订记录

版本号	修订人	修订日期	修订内容
V1.0	张辉	2015 年 10 月 23 日	修订
V1.0	张辉	2016 年 03 月 24 日	修改初始化方式
V1.0	张辉	2016 年 04 月 22 日	新增支付选项
V1.0	张辉	2016 年 07 月 20 日	简化初始化方式，不再支持分省支付，必须开通全国支付

## 目录

1. SDK调用 .....	1
1.1. 环境准备 .....	1
1.1.1. 资源导入 .....	1
1.1.2. Manifest.xml文件配置 .....	1
1.1.3. 初始化 .....	2
1.2. SDK接口.....	2
1.2.1. 支付 .....	3
1.2.2. onResume & onPause .....	4
1.2.3. 游戏退出引导（退出游戏时必须调用） .....	5

## 1. SDK调用

### 1.1. 环境准备

#### 1.1.1. 资源导入

将 sdk 压缩包中 resource 文件夹下的资源对应放入游戏工程的 assets 文件、values、libs 文件目录下，具体请参考 Demo 工程

#### 1.1.2. Manifest.xml文件配置

##### 包名

游戏包名必须以 .nearme.gamecenter 结尾

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.gcSDK.demo.nearme.gamecenter"
    android:versionCode="200"
    android:versionName="v2.0" >
```

##### 配置 Android SDK 版本

SDK 支持 Android SDK 3.0 ( Api Level 11 ) 及其以上版本

```
<uses-sdk android:minSdkVersion="11"
    android:targetSdkVersion="19"/>
```

##### 加入所需的 permission 权限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
```

##### 加入所需的组件

具体请以 Demo 中 Manifest.xml 文件实际配置为准

```
<!-- nearme game sdk config goes here -->
```

```

<meta-data android:name="debug_mode"
            android:value="false" /> <!-- 控制日志开关，发布时候设置false -->
<meta-data android:name="is_offline_game"
            android:value="true" /> <!-- true:单机游戏 false:网游 -->
<meta-data android:name="app_key"
            android:value="c5217trjnrU6g05jG8VvUFU0" /> <!--appKey，开发者后台
申请，务必使用游戏自己的appkey -->
<activity
    android:name="com.nearme.game.sdk.component.proxy.ProxyActivity"
    android:theme="@style/Theme_Dialog_Custom"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:process=":gcsdk">
</activity>
<service
    android:name="com.nearme.game.sdk.component.proxy.ProxyApiService"
    android:process=":gcsdk"
    android:priority="1000">
</service>
<!-- nearme game sdk config end -->

```

### 1.1.3. 初始化

CP 需在工程 Application 或者主进程其他位置尽早对 SDK 进行初始化, appkey 在 manifest 中配置。

```

public class DemoApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        String appSecret = "e2eCa732422245E8891F6555e999878B";
        GameCenterSDK.init(appSecret, this);
    }
}

```

## 1.2. SDK接口

SDK 中所有 API 方法都要求在主线程调用,耗时的 Api 方法 SDK 自行在非 UI 线程中运行,回调函数也将在主线程中执行, CP 不用关心线程的切换问题。GameCenterSDK 是游戏中心的 API 核心类,提供了全部的 API 方法,方法名大部分都是以 doXXX 开头.例如,游戏中心的初始化,请求登录,获取用户信息,支付,分享游戏信息等,SDK 中的所有 API 调用采用回调模式,回调的通用接口为:ApiCallback

```

public interface ApiCallback {
    /**
     * @param resultMsg
     */
    public void onSuccess(String resultMsg);
    /**
     * @param resultMsg
     * @param resultCode
     */
    public void onFailure(String resultMsg, int resultCode);
}

```

注：调用以下任何接口前，请务必保证游戏已经成功初始化。

### 1.2.1. 支付

```

GameCenterSDK.getInstance().doSinglePay(this, payInfo,
    new SinglePayCallback() {
        // add OPPO 支付成功处理逻辑~
        public void onSuccess(String resultMsg) {
            Toast.makeText(DemoActivity.this, "支付成功", Toast.LENGTH_SHORT).show();
        }
        // add OPPO 支付失败处理逻辑~
        public void onFailure(String resultMsg, int resultCode) {
            if (PayResponse.CODE_CANCEL != resultCode) {
                Toast.makeText(DemoActivity.this, "支付失败", Toast.LENGTH_SHORT).show();
            } else {
                // 取消支付处理
                Toast.makeText(DemoActivity.this, "支付取消", Toast.LENGTH_SHORT).show();
            }
        }
        /* bySelectSMSPay 为true表示回调来自于支付渠道列表选择短信支付，false表示没有
        网络等非主动选择短信支付时候的回调 */
        public void onCallCarrierPay(PayInfo payInfo, boolean bySelectSMSPay) {
            // add 运营商支付逻辑~
            Toast.makeText(DemoActivity.this, "运营商支付", Toast.LENGTH_SHORT).show();
        }
    });

```

其中 PayInfo 对象需要 CP 自己实现

```

private PayInfo createTestPayInfo(int amount){

```

```

PayInfo payInfo = new PayInfo(System.currentTimeMillis() +
    new Random().nextInt(1000) + "", "自定义字段", amount);
payInfo.setProductDesc("商品描述");
payInfo.setProductName("300符石");
// payInfo.setShowCpSmsChannel(true); // sdk支付界面是否显示运营商短信入口, true显示,
//                                     false不显示
// payInfo.setUseCachedChannel(true); // 设置是否使用上次使用过的支付方式
// 支付结果服务器回调地址, 不通过服务端回调发货的游戏可以不用填写~
payInfo.setCallbackUrl(
    "http://gamecenter.wanyol.com:8080/gamecenter/callback_test_url");
return payInfo;
}

```

#### PayInfo 属性详细说明

名称	类型	长度 ( 字符 )	是否必填	说明
order	String	100	是	订单号, 务必保证唯一
attach	String	200	否	自定义回调字段
amount	int		是	消费总金额, 单位为分
productName	String	40	是	商品名
productDesc	String	120	否	商品描述
callbackUrl	String		否	支付回调地址, 不需要服务端回调的可不填
setShowCpSmsChannel		true/false	否	在 sdk 支付界面是否显示短信支付入口, 默认不显示 ( false ), true 代表显示。
setUseCachedChannel		true/false	否	是否直接使用上一次支付所用的方式, 比如上次用微信, 这次直接跳微信支付。

- SDK 中均按分计算, 如若商品价值 5 元, 那 amount 应填写 500,
- 调用此接口应保证订单唯一, 示例中订单号由时间戳和一个随机值生成, CP 可以沿用, 也可以自己定义, 但务必保证订单号唯一
- 自定义字段为交易辅助说明字段, 建议 CP 填写此次交易相关信息
- 回调地址为 CP 服务端回调地址, 服务端接收到我方后台发出的成功回调后即可发放游戏道具

#### 1.2.2. onResume & onPause

```

@Override
protected void onResume() {
    super.onResume();
    GameCenterSDK.getInstance().onResume(this);
}

```

```
@Override
protected void onPause() {
    GameCenterSDK.getInstance().onPause();
    super.onPause();
}
```

CP 必须在**游戏主页面**调用这对接口来通知 sdk 做一些状态控制。

### 1.2.3. 游戏退出引导（退出游戏时必须调用）

```
GameCenterSDK.getInstance().onExit(Context context, GameExitCallback callback);
```

游戏准备退出时需要调用此方法，并在GameExitCallback的回调exitGame()中实现游戏真正的退出操作（比如：先进行数据存档，然后关闭游戏进程），关闭游戏进程可以参考接入demo里面使用AppUtil工具类的exitGameProcess方法完成，也可以自己实现。