

# 靠谱 SDK 接入文档

版本号	修改时间	内容	修改人
1.0	2015-9-25	初始版本	方鹏程

## 目录

1. 概述.....	5
2. 接入流程.....	5
2.1 申请应用配置 .....	5
2.2 导入资源包 .....	5
2.2.1 导入 jar 包 .....	7
2.2.2 添加 res 目录下文件.....	7
2.2.3 添加 assets 目录下文件.....	7
2.3 配置应用的 AndroidManifest.xml .....	7
2.3.1 添加权限 .....	7
2.3.2 添加 activity.....	7
2.3.3 添加 receiver.....	8
2.3.4 添加 service .....	8
2.3.5 添加 meta-data.....	9
2.4 调用授权接口 .....	10
2.5 调用登录接口 .....	10
2.6 调用支付接口 .....	10
2.7 释放 SDK 资源.....	10
2.8 在 Activity 生命周期中添加方法.....	10
3. 接口说明.....	10

3.1 授权接口.....	10
3.2 登录接口.....	11
3.3 OAuth2.0 登陆验证接口.....	12
3.4 支付接口.....	13
3.5 释放 SDK 资源接口.....	14
3.6 SDK 浮窗显示接口.....	14
3.7 SDK 浮窗关闭接口.....	14
3.8 启动 MIUI 引导页接口.....	15
3.9 是否授权成功接口.....	15
3.10 注册游戏账号注销回调对象接口.....	16
3.11 角色数据采集接口.....	16
3.12 获取当前是否登陆成功接口.....	17
3.13 注销登录接口.....	17
4. 参数类介绍.....	17
4.1 订单信息类.....	17
4.2 订单配置类.....	18
4.3 授权回调类.....	19
4.4 登陆回调类.....	20
4.5 登陆成功返回参数类.....	20
4.6 注册回调类.....	21
4.7 注册成功返回参数类.....	21
4.8 注销账号回调类.....	22

4.9 屏幕类型类.....	22
5.Web 端充值结果通知接口.....	23
5.1 功能.....	23
5.2 通知方式.....	23
5.3 请求示例.....	23
5.4 返回结果.....	24
5.5 接口参数说明.....	24
6.注意事项.....	25
7.附表 code.....	26
8.附表 PayType.....	26
9.附表 errdesc.....	26

# 1. 概述

本文档面向安卓开发者，用于指导开发者快速接入 靠谱 SDK，本 SDK 为本 SDK 用于 Android APP 嵌入用户中心和收费模块，用户中心目前支持使用靠谱论坛账号登录，收费模块支持支付宝和银行卡支付。提供多个接口给集成者调用。

## 重要提醒：

所有的 SDK 操作都必须在应用成功授权之后才能继续执行

必须把 KAOPU\_APPKEY、KAOPU\_SECRETKEY、KAOPU\_APPVERSION、KAOPU\_APPID、

KAOPU\_CHANNELINNER\_ID 五个值填写在 AndroidManifest 文件中

# 2.接入流程



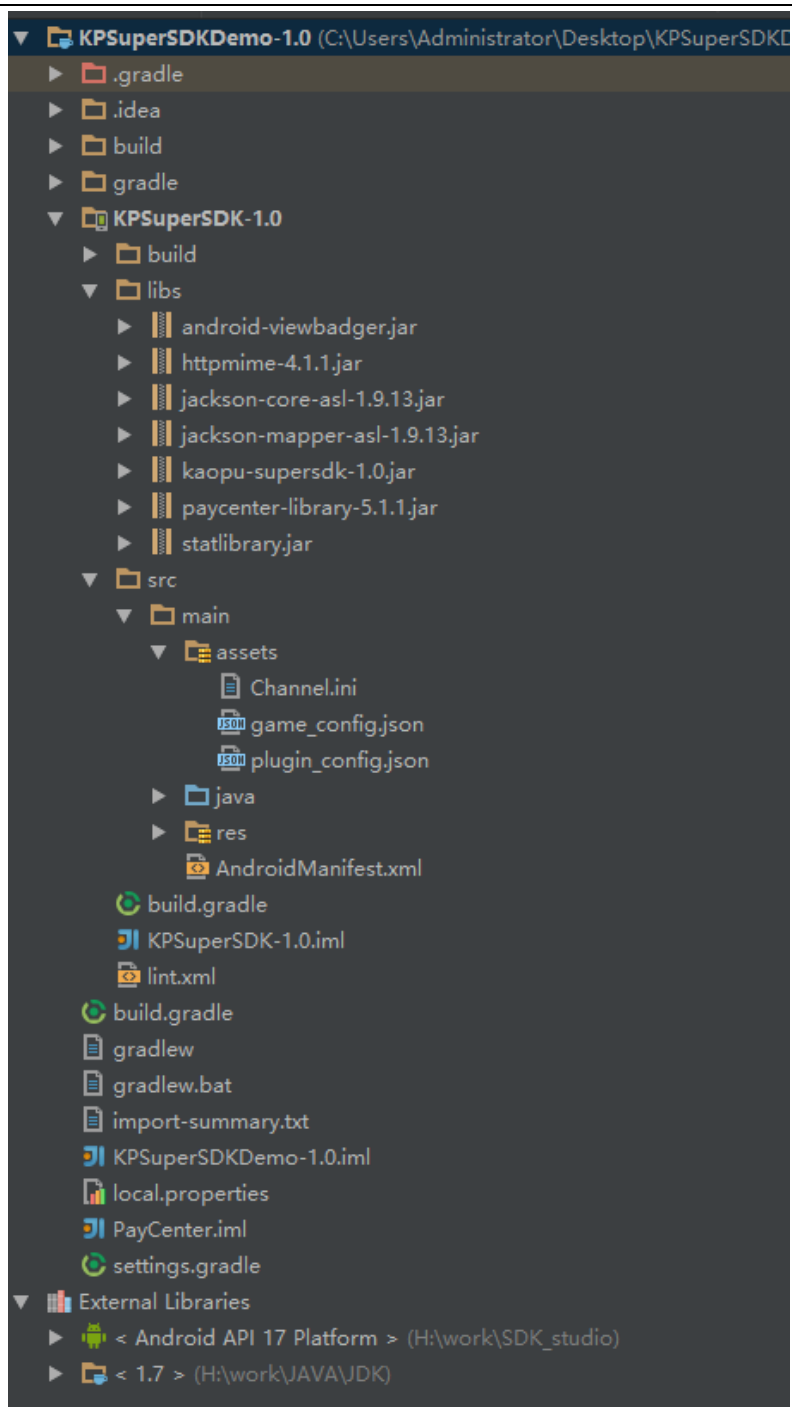
## 2.1 申请应用配置

开发者需要向靠谱SDK运营人员索取以下相应配置（[请勿直接使用demo中的参数](#)）：

KAOPU\_APPKEY、KAOPU\_SECRETKEY、KAOPU\_APPID 和 KAOPU\_APPVERSION、KAOPU\_CHANNELINNER\_ID

## 2.2 导入资源包

本 SDK 目前支持 Android2.2 及以上的系统版本，为兼容 Android 4.0 及以上的新手机，编译时请使用 Android4.2 或以上的版本，靠谱SDK 开发环境的配置流程如下（参考下图 demo开发环境配置，导入 demo 工程请以 utf-8 编码，Android4.2 导入）：



## 2.2.1 导入 jar 包

将 SDK 包内的 libs 目录下的文件（夹）放到应用工程的 libs 目录下

## 2.2.2 添加 res 目录下的文件

将 res 目录下的文件（夹）复制到应用工程 res 目录下

## 2.2.3 添加 assets 目录下的文件

将 assets 目录下的文件（夹）复制到应用工程 assets 目录下

## 2.2.4 修改 assets 目录下的 game\_config.json

此文件数据为json字符创，key 和 value 对应关系如下表所示

KEY	可否为空	描述
KP_Channel	不可以	游戏合作渠道 (请填写“kaopu”，无需修改)
ChannelKey	不可以	默认渠道值 (请填写“kaopu”，无需修改)
gameName	不可以	游戏名
screenType	不可以	屏幕方向 横屏：1 竖屏：2
fullScreen	不可以	是否全屏（全屏即：不带状态栏） 全屏：true 不全屏：false
param	可以	自定义参数

## 2.3 配置应用的 AndroidManifest.xml

### 2.3.1 添加权限

```
W<!--添加 靠谱超级 SDK 必需的权限。 -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>W
```

### 2.3.2 添加 activity

注意:必须放入<application>元素区内

```
<!--添加 靠谱超级 SDK 必需的 activity。 -->
<activity android:name="com.cyjh.pay.activity.PayCenterActivity" >
</activity>
<activity android:name="com.cyjh.pay.activity.GuideActivity" >
</activity>
<activity android:name="com.cyjh.pay.activity.PayWebActivity"
    android:configChanges="orientation|keyboardHidden|screenSize" >
</activity>
```

### 2.3.3 添加 receiver

```
<!--添加 靠谱超级 SDK 必需的 receiver。 -->
<receiver android:name="com.cyjh.pay.download.DownloadCompleteReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.DOWNLOAD_COMPLETE" />
    </intent-filter>
</receiver>
```



## 2.3.4 添加 service

```
<!--添加 靠谱超级 SDK 必需的 service。 -->
<service android:name="com.cyjh.pay.service.PushMsgService" />
```

## 2.3.5 添加 meta-data

注意：

1. 必须放入<application>元素区内：

2. ( **警告** ) 请勿直接使用 demo 中的参数，demo 参数无法作为正式参数使用，仅仅作为测试用途

```
<!--添加 靠谱 SDK 必需的 meta-data : KAOPU_APPKEY。此处 value 仅用于示例，请在 DK 平台注册申请 -->
<meta-data
    android:name="KAOPU_APPKEY"
    android:value="20004" />
<!--添加 靠谱 SDK 必需的 meta-data : KAOPU_SECRETKEY。此处 value 仅用于示例，请在 SDK 平台注册申请 -->
<meta-data
    android:name="KAOPU_SECRETKEY"
    android:value="5711DD92-7A86-4A08-BE19-DA6D782BBE28" />
<!--添加 靠谱 SDK 必需的 meta-data : KAOPU_APPID。此处 value 仅用于示例，请在 SDK 平台注册申请 -->
<meta-data
    android:name="KAOPU_APPID"
    android:value="20004001" />
<!--添加 靠谱 SDK 必需的 meta-data : KAOPU_APPVERSION。此处 value 仅用于示例，请在 SDK 平台注册申请默认为 1.0 ， 请勿修改-->
<meta-data
    android:name="KAOPU_APPVERSION"
    android:value="1.0" />
<!--添加 靠谱 SDK 必需的 meta-data : KAOPU_CHANNELINNER_ID。此处 value 仅用于示例，请在 SDK 平台注册申请,默认为 kaopu ， 请勿修改-->
<meta-data
    android:name="KAOPU_CHANNELINNER_ID"
    android:value="kaopu" />
```

## 2.4 在混淆文件中加入以下内容

添加以下内容到混淆文件中：

```
-dontpreverify
-dontwarn
-dontnote
-verbose

-libraryjars /libs/kaopu-supersdk-1.0.jar ( 请注意比对靠谱超级SDK包的名字 )

-libraryjars /libs/statlibrary.jar
-libraryjars /libs/httpmime-4.1.1.jar
-libraryjars /libs/jackson-core-asl-1.9.13.jar
-libraryjars /libs/jackson-mapper-asl-1.9.13.jar

-keepattributes Signature,*Annotation*,*EnclosingMethod*
-keepclassmembers public final enum

-keep class com.cyjh.pay.**{*;}
-keep class com.kaopu.supersdk.**{*;}
-keep class com.cyjh.statlibrary.**{*;}
-keep class org.codehaus.** { *; }

-keepnames class com.fasterxml.jackson.** { *; }
org.codehaus.jackson.annotate.JsonAutoDetect$Visibility {
public static final org.codehaus.jackson.annotate.JsonAutoDetect$Visibility *; }
```

## 2.5 在程序入口处调用授权接口（见 3.1）

## 2.6 调用登录接口（见 3.2）

## 2.7 调用支付接口（见 3.4）

## 2.8 在程序退出时调用释放 SDK 资源接口（见 3.5）

## 2.9 在程序 Activity 的生命周期中添加相应方法（见 3.6 & 3.7）

# 3 接口说明

## 3.1 授权接口

此接口必须在APP刚启动时(必须在登录之前)调用，用于给当前APP授权，授权成功后才可以进行登录和支付。调用方法如下：

```
KPSuperSDK.auth(Activity activity, KPAuthCallBack callBack);
```

接口所需参数：

参数	可否为空	描述
Activity activity	不可以	上下文

<b>KPAuthCallback callback</b>	不可以	授权回调对象，用于处理授权结果，若传null，则表示不处理， <b>具体详见4.3</b>
------------------------------------	-----	---

## 3.2 登陆接口

此接口请在**授权完成后的登录操作**调用。（授权相关请查看 3.1）。此接口调用方法如下：

```
KPSuperSDK.login(Activity activity, KPLoginCallBack callback)
```

接口所需参数：

参数	可否为空	说明
Activity activity	不可以	上下文
KPLoginCallBack callback	不可以	处理登录结果的回调, <b>具体详见4.4</b>

## 3.3 Oauth2.0 登陆验证接口

### 3.3.1 获取用户身份验证有效性的 url 地址

接口地址：	<a href="http://sdk.geturl.kpzs.com/api/UserAuthUrl?tag={0}&amp;tagid={1}&amp;appid={2}&amp;version={3}&amp;imei={4}&amp;channelkey{5}&amp;r={6}&amp;sign={7}">http://sdk.geturl.kpzs.com/api/UserAuthUrl?tag={0}&amp;tagid={1}&amp;appid={2}&amp;version={3}&amp;imei={4}&amp;channelkey{5}&amp;r={6}&amp;sign={7}</a>
四组随机秘	18257284-7F5D-348D-AB09-299E5B7DD997

钥:	655A957D-157D-7C21-E3A7-9CAAF835318 F467CA93-D550-346D-6BCB-173995F7C83A BD32817A-99F9-2E26-5B33-15208F7B360A
调用方式:	Get
入口签名:	<b>sign: 所有请求参数值+4组随机密钥之一,按字母顺序升序拼接生成MD5</b>
入口参数	tag: 99999 tagid: 53F3-5F82-47BD-9FC3-F5FFEB653C8 appid:99999001 version: 2.0 imei: 0722100523 r: 1 sign: 790df0eabb61eda62387c3a834a7  各部分表示意义为: tag: 合作标识 key (meta-data : KAOPU_APPKEY , <a href="#">详见 2.3.5</a> ) tagid: 合作方 ID(meta-data : KAOPU_SECRETKEY , <a href="#">详见 2.3.5</a> ) appid: 合作者应用 ID(meta-data : KAOPU_APPID , <a href="#">详见 2.3.5</a> ) version: 当前 sdk 版本 (在登陆回调中返回, <a href="#">详见 4.4</a> ) imei: 手机 IMEI (在登陆回调中返回, <a href="#">详见 4.4</a> ) channelkey 渠道标识 (在登陆回调中返回, <a href="#">详见 4.4</a> ) r: sign 中所使用的随机密钥组 ID。 sign: 所有请求参数+4 组随机密钥组值, 按字母升序排序生成 md5
返回值签名:	sign: code + tag +imei + 4组随机密钥之一
返回值:	返回 json 字符串 <pre>{   "code": 1,   "msg": "获取成功",   "sign": "0eabb61eda62387c3a834a ",   "r": 1,   "data":     {       url:"http://sdk.kpzs.com/api/checkxx"     } }</pre>

	<p>说明:</p> <p>code:数字, 字典: 0 失败 1 成功获取</p> <p>msg:返回的文字提示信息。</p> <p>sign:签名</p> <p>r:数字签名的随机密钥串的 ID</p> <p>data: 请求成功有值, 失败无值</p>
--	---

### 3.3.2 用户身份有效性验证

**参数获取** 详见 [4.4 登陆回调类](#) 和 [4.5登陆回调参数类](#)

接口地址:	<a href="#">{url地址获取详见 3.3.1}?devicetype={0}&amp;imei={1}&amp;r={2}&amp;tag={3}&amp;tagid={4}&amp;appid={5}&amp;channelkey={6}&amp;openid={7}&amp;token={8}&amp;sign={9}</a>
四组随机密钥:	18257284-7F5D-348D-AB09-299E5B7DD997 655A957D-157D-7C21-E3A7-9CAAF835318 F467CA93-D550-346D-6BCB-173995F7C83A BD32817A-99F9-2E26-5B33-15208F7B360A
调用方式:	Get
入口签名:	<b>sign:</b> 所有请求参数值+4组随机密钥之一,按字母顺序升序拼接生成MD5( 详见: <a href="#">4.5登陆回调参数类</a> , 已返回, 无需自己拼接 )
入口参数	<pre> “devicetype”:“android”, “imei”:“357139050149462”, “sign”:“f39126a72fd4a9b60f67e164fea17ad5”, “r”: 1 “tagid”:“151E635D-1D7C-4A27-942B-684F4D1A025” “tag”: 21111 “appid”:10001001 “channelkey: “kaopu”, “openid”:“abc12312e”, “token”:“ef45123de”, </pre> <p>各部分表示意义为: 详见</p> <p>devicetype: 设备类型 android ios</p> <p>imei: 手机标识</p> <p>sign: 所有请求参数+4 组随机密钥组值, 按字母升序排序生成 md5</p>

	r: sign 中所使用的随机密钥组 ID。 tag: 合作标识 key tagid: 合作方 ID appid: 合作者应用 ID channelkey: 渠道应用标识 openid: 用户 ID token: 用户通信 token
返回值签名:	sign: code + tag +imei+token+4组随机密钥之一
返回值:	返回 json 字符串 <pre>{   "code": "0",   "msg": "错误消息",   "sign": "数字签名",   "r": "数字签名的随机密钥串的 ID",   "data": null }</pre> 说明: code:数字, 字典: 1 处理成功 0 处理失败 201 Token 失效(过期) 202 Token 验证失败 msg:返回的文字信息。 sign:签名 r:数字签名的随机密钥串的 ID data: null

## 3.4 支付接口

此接口请在[授权完成后](#)调用。(授权相关请查看 3.1)。此接口调用方法如下:

```
KPSuperSDK.pay(Activity activity, PayParams payParams, KPPayCallBack payCallBack);
```

接口所需参数:

参数	可否为空	说明
Activity activity	不可以	上下文
PayParams payParams	不可以	订单信息
KPPayCallBack payCallBack	不可以	处理支付结果的回调

### 3.5 释放 SDK 资源接口 ( 请务必接入 )

此接口请在程序退出的时候调用，此接口调用方法如下：

```
KPSuperSDK.release();
```

### 3.6 SDK 浮窗显示接口 ( 请务必接入 )

此接口需要保证在所有 Activity 生命周期里的 onResume()里调用，此接口调用方法如下：

```
KPSuperSDK.showFloatView(Context context);
```

### 3.7 SDK 浮窗关闭接口 ( 请务必接入 )

此接口需要保证在所有 Activity 生命周期里的 onStop()里调用，此接口调用方法如下：



```
KPSuperSDK.closeFloatView(Context context);
```

### 3.8 注册游戏账号注销回调对象接口

此接口请于[授权成功之后，登陆之前调用](#)，用于当用户点击 SDK 切换账号按钮 和 调用注销接口时回调。调用方法如下：

```
KPSuperSDK.registerLogoutCallBack(KPLogoutCallBack mLogoutCallBack);
```

接口所需参数：

参数	说明
KPLogoutCallBack	注销回调
mLogoutCallBack	详见： <a href="#">4.5 注销账号回调</a>

### 3.9 角色数据采集接口

此接口必须在[登录成功且角色创建成功后](#)调用，调用方法如下：

```
KPSuperSDK.setUserGameRole(Context context, String serviceName, String rolename, String uopenid, int roleLevel);
```

接口所需参数：

参数	可否为空	说明
Context	不可以	上下文
String serviceName	不可以	角色所在区服名称
String rolename	不可以	角色名称
String uopenid	不可以	游戏方对外用户 ID
int roleLevel	不可以	角色等级

### 3.10 获取当前是否登录成功接口

此接口用于判断当前是否登录成功，调用方法如下：

```
KPSuperSDK.isLogin();
```

若返回 true 则表示已登录。反之未登录。

### 3.11 注销登录 接口

此接口用于手动退出登录，调用方法如下：

```
KPSuperSDK.logoutAccount();
```

## 4 参数类介绍

### 4.1 订单信息类 PayParams

PayOrder 为调用支付接口需要传递的参数，以下几个字段需要调用者提供（必填）：

字段	可否为空	说明
Amount	不可以	充值金额
Gamename	不可以	游戏名称
GameServer	不可以	充值服务器
Rolename	不可以	充值角色名称
Orderid	不可以	唯一订单号
Currencyname	可以	虚拟货币名称
Proportion	可以	<b>RMB:</b> 虚拟货币的比例
isCustomPrice	可以	是否自定义价格
CustomText	可以	自定义的金额描述

使用示例如下:

```
PayParams payParams = new PayParams();  
payParams.setAmount(300); // 充值金额  
payParams.setGamename("刀塔传奇"); // 充值游戏  
payParams.setGameserver("137区"); // 充值游戏服务器  
payParams.setRolename("花花宝贝"); // 充值角色名称  
payParams.setOrderid("1112323245566") //一订单号
```

注意：在不知道比例的情况下，可设置 `isCustomPrice` 为 `true`，SDK将显示自定义的 `CustomText`。

参数验证情况：

- 知道比例的情况下，即设置 `isCustomPrice` 为 `false` 时，`Currencyname` 和 `Proportion` 不可以为空，若传空值参数验证将不通过并提示参数不合法，使用示例如下：

```
PayParams payParams = new PayParams();  
payParams.setCurrencyname("钻石"); // 虚拟货币名称  
payParams.setProportion(10); // 10 表示比例为 1RMB:10 虚拟货币
```

- 不知道比例的情况下，即设置 `isCustomPrice` 为 `true` 时，`Currencyname` 和 `Proportion` 可以为空，SDK 将不对这 2 个参数做验证。使用示例如下：

```
PayParams payParams = new PayParams();  
payParams.setCustomPrice(true);  
payParams.setCustomText("0.01 元兑换 500 钻石");
```

## 4.2 授权回调类 KPAuthCallBack

说明：授权接口的回调对象。以下为接口中提供的方法：

方法	说明
onAuthSuccess()	授权成功
onAuthFailed()	授权失败

## 4.3 登陆回调类 KPLoginCallBack

说明：登陆接口的回调对象。以下为接口中提供的方法：

方法	说明
onLoginSuccess(UserInfo userInfo)	登录成功
onLoginFailed()	登录失败
OnLoginCanceled()	登录取消

## 4.4 登陆成功返回参数类 UserInfo

说明：登陆成功后通过登陆回调接口返回的对象。以下为对象属性：

字段	说明
String username	登录用户名
String userid	用户id
String openid	OAuth登录验证中的参数 openid
String iconurl	登录用户头像地址
String description	登录描述
String token	OAuth登录验证中的参数 token
String tag	OAuth登录验证中的参数 tag
String tagid	1. OAuth登录验证中的参数 tagid 2. 获取 <a href="#">用户身份验证有效性的url地址 (3.3.1)</a> 接口中的额参数tagid
String appid	1. OAuth登录验证中的参数 appid 2. 获取 <a href="#">用户身份验证有效性的url地址 (3.3.1)</a> 接口中的额参数appid
String channelKey	1. OAuth登录验证中的参数 channelKey 2. 获取 <a href="#">用户身份验证有效性的url地址 (3.3.1)</a> 接口中的额参数 channelKey

String r	OAuth登录验证中的参数 r
String devicetype	OAuth登录验证中的参数 devicetype
String imei	1. OAuth登录验证中的参数 imei 2. 获取 <a href="#">用户身份验证有效性的url地址 (3.3.1)</a> 接口中的参数 imei
String sign	OAuth登录验证中的参数 sign
String version	获取 <a href="#">用户身份验证有效性的url地址 (3.3.1)</a> 接口中的参数 version

## 4.5 注销账号回调类 KPLLogoutCallBack

说明：注销时的回调接口（[此对象注册方法详见：3.10 注册游戏账号注销回调对象接口](#)）以下为接口

提供的方法：

方法	说明
onSwitch()	切换账号时执行（ <a href="#">此时自动弹出SDK登陆窗口</a> ）
onLogout()	退出账号时执行，包括以下两种情况： （ <a href="#">此时不会弹出SDK登陆窗口</a> ） 1. 手动调用函数 <code>KPSuperSDK.logoutAccount()</code> 2. 账户身份信息存在时间超时，需要重新登录

请在 `onSwitch` 和 `onLogout` 中调用当前游戏的注销逻辑。

## 4.6 支付回调类 KPPayCallBack

说明：支付接口的回调对象。以下为接口中提供的方法：

方法	说明
onPaySuccess()	支付成功
onPayFailed()	支付失败
OnPayCancel()	除了以上两种情况外，退出支付页面，就会调用

## 5 Web 端充值结果通知接口

### 5.1 功能

用户在靠谱 SDK 平台进行充值后服务器需要将充值结果通知游戏服务器

### 5.2 通知方式（HTTP GET）

接收充值数据的接口地址（游戏方提供）如：

<http://xxx.xxx.xxx/xxx.aspx>



## 5.3 请求示例

```
http://xxx.xxx.xxx/xxx.aspx?username=kaopu&kpordernum=12345679&ywordernu  
m=12345&status=1&paytype=1&amount=1222&errdesc=1000&paytime=201211292  
34817&gamename=abcd&sign=1233333&gameserver=XXXX
```

## 5.4 返回结果（JSON 格式）

```
{  
  "code": "1000",  
  "msg": "错误信息",  
  "sign": "123",  
}
```

code: 参考附表 **code**

msg: 系统异常描述信息

sign: 由 (code 值|signkey) MD5, signkey 指的是: AndroidManifest.xml 配置文件中 **KAOPU\_SECRETKEY** 对应的值 (详见 2.3.5)

## 5.5 接口参数说明

参数名	参数类型	参数说明	示例
username	string	充值用户名 (utf8 编码传输)	kpzs
kpordernum	string	订单 ID, 靠谱订单系统唯一号	20141125123012230001
ywordernum	string	订单 ID, 游戏订单系统唯一号	由调用方提供, 会原样返回

status	string	订单状态,1 为成功,其他为失败	1
paytype	string	充值类型, 详见附表PayType	1
amount	int	成功充值金额, 单位为分	10000
gameserver	string	游戏区服	
errdesc	string	充值失败错误码, 成功为空。 见附表 errdesc	
paytime	string	充值成功时间 ,yyyyMMddHHmmss	20121129234817
gamename	String	游戏名称(utf-8 编码)	西游

sign	string	<p>所有参数(除 sign)的 MD5</p> <p>签名拼串: username 值</p> <p> kwordernum 值</p> <p> ywordernum 值  status</p> <p>值  paytype 值  amount</p> <p>值 gameserver 值 </p> <p>errdesc 值  paytime 值 </p> <p>gamename 值 signkey</p> <p>值</p>	<p><b>(警告)</b>: 参数拼接是,</p> <p>“ ”一定要拼接进去, 每</p> <p>两个参数之间都需要一个</p> <p>“ ”进行拼接)</p> <p>验证说明:合作方接收到我</p> <p>们的数据后根据 sign 的格</p> <p>式进行拼装好, 然后进行</p> <p>MD5 后与我们传过来的</p> <p>sign 做比较, 确保两个值是</p> <p>一致的 (拼装时不要把 sign</p> <p>拼装进去)</p> <p><b>signkey</b> :<a href="#">AndroidManifest.xml</a></p> <p>配置文件中 <b>KAOPU_SECRETKEY</b></p> <p>对应的值 (详见 2.3.5)</p>
------	--------	---	--

## 6 注意事项

### 6.1

所有 SDK 操作 都需要在调用授权接口成功以后才能继续执行

## 6.2

SDK 大部分窗口都是浮窗，在小米系列手机测试时，将启动引导页，需要点击引导页打开应用详情页并打开对应的浮窗功能才能正常显示。

## 7 附表 code

类别 ID	类别名称
1000	处理成功
1002	验签失败
1003	订单号不存在
1004	参数异常
1005	系统异常
1006	用户名不存在
1007	游戏名不存在
1008	区服不存在
1009	金额不正确

## 8 附表 PayType

类别 ID	类别名称
1	支付宝
2	财付通
3	移动充值卡
4	电信充值卡
5	联通充值卡
6	银行卡
7	易宝支付

## 9 附表 errdesc

错误码	错误描述
0	网络连接失败
1	销卡成功，订单失败
1000	0 未知错误
1002	本张卡密您提交过于频繁，请您稍后再试
1003	不支持的卡类型（比如电信地方卡）
1004	密码错误或充值卡无效
1006	充值卡无效
1007	卡内余额不足
1008	余额卡过期（有效期 1 个月）
101	md5 验证失败
1010	此卡正在处理中
102	订单号重复
103	恶意用户
104	序列号，密码简单验证失败
105	密码正在处理中
106	系统繁忙，暂停提交
107	多次支付时卡内余额不足
109	des 解密失败
200	请求成功（非订单支付成功）
2005	此卡已使用
2006	卡密在系统处理中
2007	该卡为假卡
2008	该卡种正在维护
2009	浙江省移动维护
201	证书验证失败
2010	江苏省移动维护
2011	福建省移动维护
2012	辽宁省移动维护
2013	该卡已被锁定
2014	系统繁忙，请稍后再试
3001	卡不存在
3002	卡已使用过
3003	卡已作废
3004	卡已冻结
3005	卡未激活
3006	密码不正确
3007	卡正在处理中

## 靠谱 SDK 接入文档

3101	系统错误
3102	卡已过期
4000	系统异常
5001	订单支付失败