# 华为TV游戏中心SDK开发指导书

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 说明 |
| V1.0 | 2015-2-20 | 初稿完成 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目录

[华为TV游戏中心SDK开发指导书 1](#_Toc415160699)

[1 概述 4](#_Toc415160700)

[1.1 SDK接入一般业务流程 4](#_Toc415160701)

[2 接入SDK前的准备工作 5](#_Toc415160702)

[2.1 登录华为开发者联盟 5](#_Toc415160703)

[2.2 应用列表 6](#_Toc415160704)

[2.3 添加支付权益 6](#_Toc415160705)

[2.4 设置回调地址 9](#_Toc415160706)

[3 集成步骤 10](#_Toc415160707)

[3.1 导入jar包 10](#_Toc415160708)

[3.2 导入assets资源 10](#_Toc415160709)

[3.3 导入res资源 11](#_Toc415160710)

[3.4 配置 AndroidManifest.xml 11](#_Toc415160711)

[3.5 编写代码 12](#_Toc415160712)

[3.5.1 账号登录 12](#_Toc415160713)

[3.5.2 退出帐号监听 13](#_Toc415160714)

[3.5.3 调起支付 14](#_Toc415160715)

[4 SDK接口描述 14](#_Toc415160716)

[4.1 华为帐号接口 14](#_Toc415160717)

[4.1.1 设置登录代理 14](#_Toc415160718)

[4.1.2 登录 15](#_Toc415160719)

[4.1.3 清除本地用户登录信息 16](#_Toc415160720)

[4.1.4 验证accesstoken请求 17](#_Toc415160721)

[4.1.5 释放登录适配器对应的资源 18](#_Toc415160722)

[4.2 华为支付接口 19](#_Toc415160723)

[4.2.1 订单数据生成 19](#_Toc415160724)

[4.2.2 调用安全支付 21](#_Toc415160725)

[4.2.3 支付结果获取和处理 22](#_Toc415160726)

[4.2.4 支付结果返回码类表 22](#_Toc415160727)

[4.2.5 RSA签名和验签 23](#_Toc415160728)

[5 代码混淆 24](#_Toc415160729)

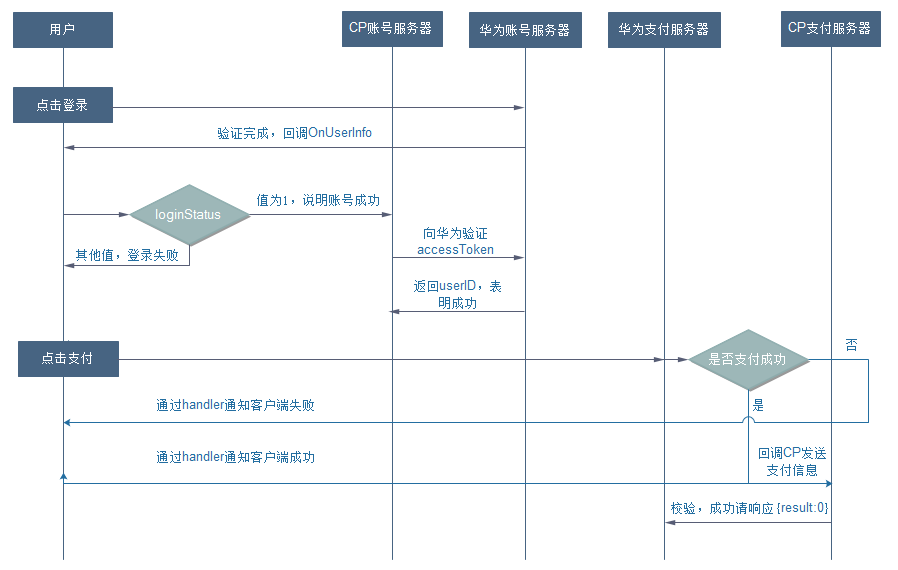
# 概述

华为游戏中心SDK向开发者提供华为帐号、华为支付。

|  |  |
| --- | --- |
| **功能** | **描述** |
| 华为帐号 | 提供华为帐号统一注册、登录功能 |
| 华为支付 | 支付Android平台SDK为第三方应用提供了支付API调用服务，使第三方客户端无需了解复杂的验证API调用过程。华为支付服务为SDK + APK 方式进行运作，当设备不存在华为安全支付服务APK时，将会在客户端进行下载操作。因此，应用需要集成支付SDK，在使用时跨进程调用支付APK来完成实际操作 |

本文介绍接入华为TV游戏中心SDK的流程。

## SDK接入一般业务流程



# 接入SDK前的准备工作

通过该过程，能够获取开发过程中需要参数，通常该部分是商务操作的，技术只需要关注2.5

## 登录华为开发者联盟

如果您登录成功后，进入会员中心，将跳转到以下页面



点击上图红框中的“创建移动应用”按钮，即可开始创建应用。

在如下的“创建应用”界面，因为还没把包做好，所以只需要填写几个基本信息（如图中的几项就行了），记得取消红框中的勾选，然后点击最下面的C:\Users\h00193325\AppData\Roaming\Tencent\Users\805526561\QQ\WinTemp\RichOle\@%Q2)M){1U~4E$DJY$QYBBV.jpg将应用保存为草稿，等应用开发好了再填写其他信息。



**温馨提示：如果集成华为支付SDK进行应用内收费，则APK包名后缀必须为“.HWTV.HUAWEI”（字母为全部大写或全部小写），应用包名自动从APK中获取，后续不能修改。**

## 应用列表

应用创建成功后，会自动跳转到“应用管理”列表页面，在“应用管理”中的应用列表中可以看到所有的应用：



## 添加支付权益

添加支付权益需要先完善银行卡信息。

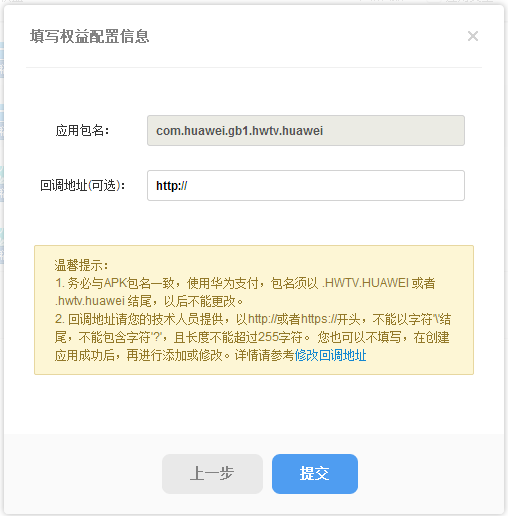
银行卡信息填写完成后，便开始添加支付权益：



点击红框中的符号C:\Users\h00193325\AppData\Roaming\Tencent\Users\805526561\QQ\WinTemp\RichOle\%A0%_ME~BX(5{16RRTBT{LU.jpg，将弹出权益添加界面



选中支付，点击下一步：



**切记，包名无法修改**，所以之前提交的应用决不能写错。

**为了支付安全可靠性考虑务必添加回调地址**，点击提交：



红框中有你需要的所有参数，点击后面的复制将参数保存下来，按照下面表格的方式存放，然后发给技术人员，切勿弄错。对应关系如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数值** | **对应图中** | **值** |
| APP\_ID | 联盟为应用分配的应用ID | App ID |  |
| PAY\_ID | 支付ID | 支付ID(原昵称) |  |
| PAY\_RSA\_PRIVATE | 支付私钥 | 支付私钥 |  |
| PAY\_RSA\_PUBLIC | 支付公钥 | 支付公钥 |  |

点击关闭，进入列表页面：



如果刚才你错过了获取参数，可以参考图中两种方式获取参数。

**方式一：**

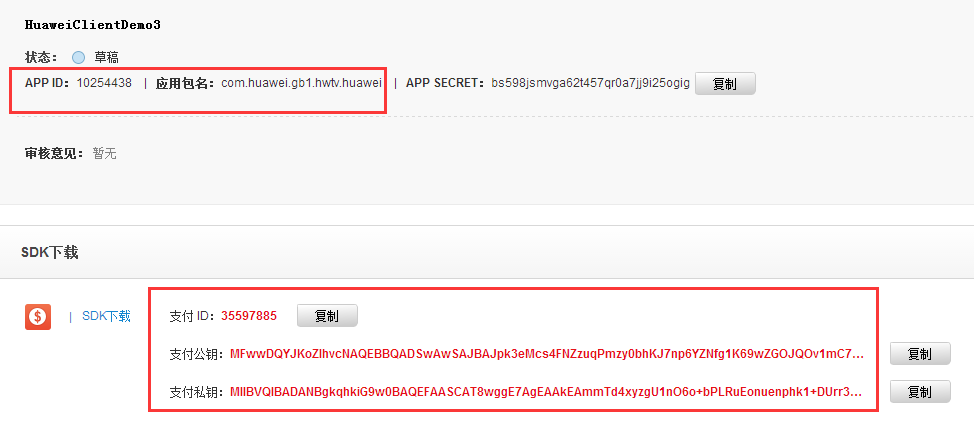
点击支付权益按钮C:\Users\h00193325\AppData\Roaming\Tencent\Users\805526561\QQ\WinTemp\RichOle\AR~9919M~B(9WAOM(CIHS{G.jpg，进入支付详情页面：



所有的参数都在红框中，如何对应，看上面表格吧。

**方式二：**

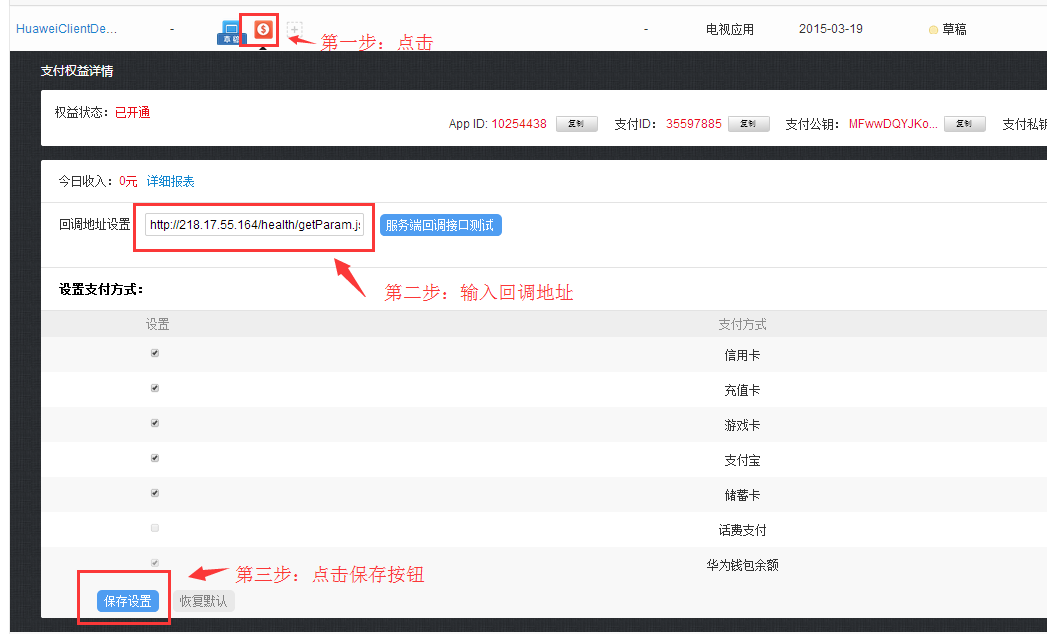
点击应用名称，进入应用详情页面：



所有的参数都在红框中，如何对应，看上面表格吧。

## 设置回调地址

在应用列表页面，找到你的应用，点击支付权益图标C:\Users\h00193325\AppData\Roaming\Tencent\Users\805526561\QQ\WinTemp\RichOle\X(S$(N[R4FE)I0]1@Y%`9GH.jpg，打开支付详情：



按照红框中的1、2、3操作，便可设置回调地址了。其中回调地址中不能够存在“?”，因为华为禁止回调中有GET参数，记得提醒你们的技术人员。

# 集成步骤

## 导入jar包

将demo下的所有libs下的jar拷入项目.

具体操作方式需要根据ADT版本来区分：

ADT20以下：放到libs或lib，并在build path中添加project jars

ADT20以上：只能放到libs，因为ADT会默认读取libs下的jar

## 导入assets资源

将assets目录下的所有文档拷贝到您工程目录对应的assets目录内，包括：

HuaweiPayService.apk：华为支付服务APK

## 导入res资源

把SDK文件夹中的res文件拷贝到业务自身的res中，帐号SDK的资源前缀名为ql\_和xh\_，在拷贝的时候一般不会出现覆盖业务自身的情况，但是建议还是拷贝之前先核对下。

【注】目标工程中尽量不要使用ql\_、xh\_开头的资源，以免同SDK中的资源冲突。

## 配置 AndroidManifest.xml

* **配置权限**

在<manifest></manifest>里面加入权限声明：

|  |
| --- |
| <uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />  <uses-permission android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />  <uses-permission android:name=*"android.permission.CHANGE\_NETWORK\_STATE"* />  <uses-permission android:name=*"android.permission.GET\_ACCOUNTS"* />  <uses-permission android:name=*"android.permission.GET\_TASKS"* />  <uses-permission android:name=*"android.permission.INTERNET"* />  <uses-permission android:name=*"android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"* />  <uses-permission android:name=*"android.permission.READ\_PHONE\_STATE"* />  <uses-permission android:name=*"android.permission.SEND\_SMS"* />  <uses-permission android:name=*"android.permission.SYSTEM\_ALERT\_WINDOW"* />  <uses-permission android:name=*"android.permission.SYSTEM\_OVERLAY\_WINDOW"* />  <uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"* />  <uses-permission android:name=*"android.permission.ACCESS\_DOWNLOAD\_MANAGER"* />  <uses-permission android:name=*"com.huawei.hwid.permission.ACCESS"* />  <uses-permission android:name=*"com.huawei.appmarket.provider.readPermission"*/> |

* 配置**Activity**

在<application></application>里面加入Activity与Receiver声明：

|  |
| --- |
| <!-- 账号依赖的Activity声明 -->  <activity  android:name=*"com.huawei.hwid.openapi.auth.dump.OpenDumpActivity"*  android:configChanges=*"orientation|screenSize"*  android:exported=*"false"*  android:theme=*"@android:style/Theme.Translucent"* />  <activity  android:name=*"com.huawei.hwid.openapi.quicklogin.ui.QuickLoginActivity"*  android:configChanges=*"orientation|screenSize"*  android:exported=*"false"*  android:theme=*"@style/Theme.quicklogin"* >  <intent-filter>  <action android:name=*"com.huawei.hwid.opensdk.ACTION\_QUICKLOGIN\_SMS"* />  <category android:name=*"android.intent.category.DEFAULT"* />  </intent-filter>  </activity> |

## 编写代码

### 账号登录

登录代理接口描述见4.1.2节

**例**（MainActivity.java的onClick方法）：

|  |
| --- |
| Bundle loginBundle = **new** Bundle();  // 为登录设置代理  OpenHwID.*setLoginProxy*(MainActivity.**this**, PartnerConfig.***applicationID***,  **new** LoginCallBack(), loginBundle);  OpenHwID.*login*(**new** Bundle()); |

登录完成后会主动调用代理中的设置的回调类LoginCallBack的onUserInfo方法，

需要写成如下方式：



### 退出帐号监听

**接口描述**

用户如果在设置->账户 中退出账户 会发送退出的广播。如果想接收这个广播请增加以下监听：

|  |  |
| --- | --- |
| 广播类型 | 全局广播 |
| 广播Intent | action: com.huawei.hwid.ACTION\_PREPARE\_LOGOUT\_ACCOUNT |

在AndroidManifest.xml中注册广播接收器

<!-- 切换帐号的广播，由开发者自行实现逻辑，并修改package路径 -->

<receiver

android:name=*"com.huawei.gb.huawei.ChangeUserReceiver"*>

<intent-filter>

<action android:name=*"com.huawei.hwid.ACTION\_PREPARE\_LOGOUT\_ACCOUNT"* />

</intent-filter>

</receiver>

广播接收*ChangeUserReceiver*需要CP替换成自己的Receiver；

|  |
| --- |
| **public** **class** ChangeUserReceiver **extends** BroadcastReceiver {  **private** **static** **final** String ***TAG*** = ChangeUserReceiver.**class**.getSimpleName();  @Override  **public** **void** onReceive(Context context, Intent intent) {  String action = intent.getAction();  DebugConfig.*d*(***TAG***, "onReceive action=" + action);  String LOGOUT\_ACTION = "com.huawei.hwid.ACTION\_PREPARE\_LOGOUT\_ACCOUNT";  **if** (LOGOUT\_ACTION.equals(action)) {  //账号切换需要删除之前保存的用户信息。  //cp自己实现  }  }  } |

### 调起支付

前提条件：如有使用华为账号，则华为帐号必须完成登录。

**例**（GameActivity.java的startPay方法）：

|  |
| --- |
| /\*\*  \* 开始支付  \*/  payHelper.startPay(activity, payInfo, handler); |

# SDK接口描述

## 华为帐号接口

### 设置登录代理

* 接口描述

设置登录的代理，为登录接口提供处理参数

* 接口原型

|  |
| --- |
| public static void setLoginProxy(Activity activity, String clientId, IHwIDCallBack callBack,Bundle bundle) |

参数说明

|  |  |  |
| --- | --- | --- |
| **参数名称** | **描述** | **备注** |
| activity | 业务的当前的Activity | 此参数必须携带 |
| clientId | 注册应用时获得的App ID | 业务在开发者联盟上注册的appId |
| callBack | 获取用户信息的回调 | 当获取用户完成（成功或者失败）后，会调用其中的接口。 |
| bundle | 扩展参数 | 该参数不赋值，或者填写为null，则OpenSDK取默认值，详见bundle说明。 |

### 登录

* 接口描述

当用户设置了登录代理后，可使用该接口进行登录：要求游戏每次启动时都调用此函数；对于已登录过的用户，会对已获取的accesstoken进行有效性认证，如果认证通过，则直接返回用户信息；如果认证失败，将跳转到登录界面重新登录，华为账号会处理此逻辑，不需要CP做处理。

当未设置登录代理时，调用该接口内部不执行任何动作。

* 接口原型

|  |
| --- |
| public static void login(Bundle bd) |

* 参数说明

无

* 使用示例

|  |
| --- |
| /\*\*  \* 对login按钮进行响应  \*/  @Override  **public** **void** onClick(View view) {  **int** id = view.getId();  **switch** (id) {  // 点击登录按钮  **case** R.id.*login*:  Bundle loginBundle = **new** Bundle();  Log.*d*("shixu", "login");  // 为登录设置代理  OpenHwID.*setLoginProxy*(MainActivity.**this**, PartnerConfig.***applicationID***,  **new** LoginCallBack(), loginBundle);  OpenHwID.*login*(**new** Bundle());  **break**;  **break**;  **default**:  **break**;  }  } |

【**备注】：**

1. 登录成功后会回调4.1.1的回调接口，业务获取用户信息以及accessToken
2. **商户可以保存用户信息，此时请务必使用userID作为唯一身份识别**

### 清除本地用户登录信息

* 接口描述

清除本地用户登录信息。并不会注销系统华为帐号。游戏下次进入时，调login接口会自动以系统华为帐号完成登录。

**由于华为是单账号系统，非特别情况，不要调用此函数**

* 接口原型

|  |
| --- |
| public static void logout() |

* 参数说明

无

* 使用示例

|  |
| --- |
| B\_logout.setOnClickListener(**new** OnClickListener() {  @Override  **public** **void** onClick(View v) {  hwIDOpenSDKInstance.logout();  valueTextView.setText("请登录");  }  }); |

### 验证accesstoken请求

华为帐号服务器提供给第三方服务器的accesstoken认证接口。

* 请求
* 请求协议：HTTPS
* 请求方法： POST或者GET
* 请求地址：https://api.vmall.com/rest.php
* 请求Body参数：
* nsp\_svc=OpenUP.User.getInfo&nsp\_ts=1358237366&access\_token=mF\_9.B5f-4.1JqM
* 注：nsp\_ts  服务请求时间戳，自GMT 时间 1970-1-1 0:0:0至今的秒数。PHP中，通过time()函数获取。 如果传入的时间与服务器时间相差5分钟以上，服务器可能会拒绝请求。nsp\_svc写死为OpenUP.User.getInfo
* 返回值

返回值（json编码格式）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **可选** | **类型** | **长度** | **描述信息** |
| userID | M | String | 20 | 用户ID |
| userName | M | String | 50 | 用户名；优先显示昵称；没有昵称则显示帐号（隐藏部分字符） |
| languageCode | M | String | 4 | 语言代码(参考附件ISO 639) |
| userState | O | Int |  | 用户验证状态0:未验证1:已验证（该参数预留，目前不作为是否合法登录的依据） |
| userValidStatus | O | Int |  | 用户有效状态 1 正常  2 暂停  3 销户（该参数预留，目前不作为是否合法登录的依据） |

返回值示例：

HTTP/1.1 200 OK

Date: Tue, 15 Jan 2013 08:09:57 GMT

Pragma: no-cache

Cache-Control: no-cache

Expires: Thu, 01 Jan 1970 00:00:00 GMT

Content-Type: text/plain; charset=utf-8

{"gender":2,"languageCode":"zh","userID":"1727","userName":"chenshu","userState":1,"userValidStatus":1}

* 异常

HTTP/1.1 200 OK

Date: Tue, 15 Jan 2013 08:12:38 GMT

Pragma: no-cache

Cache-Control: no-cache

Expires: Thu, 01 Jan 1970 00:00:00 GMT

Content-Type: text/plain; charset=utf-8

NSP\_STATUS: 109

{"error":"nsp\_ts error"}

注：NSP\_STATUS为错误码

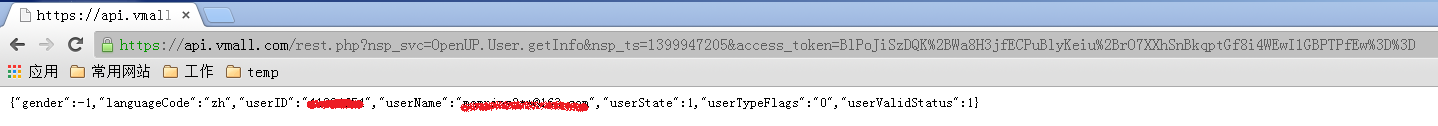
* **重要提醒**

1、在验证请求时，请使用https协议；

2、在填写access\_token参数值前，请对access\_token参数值进行编码并替换加号，**例如,用java**编码方法为:java.net.URLEncoder.encode(access\_token, "utf-8")，然后将符号“+”使用replace(“+”, “%2B”)进行替换；

3、返回值，只要存在userID就说明校验成功；

* 样例



浏览器地址栏内容样例：

https://api.vmall.com/rest.php?nsp\_svc=OpenUP.User.getInfo&nsp\_ts=1407761554&access\_token=BlSJakCh5r6wdDVWF23unifdpouZZBnABwUuqrfjfNSoqFxxX7u7qJHrr9bDhQ%3D%3D

### 释放登录适配器对应的资源

* 接口描述

释放登录适配器对应保存的资源，因该静态类中保存了activity等相关资源，需显示释放内部静态变量。当未设置适配器时，调用该接口内部不执行任何动作。

* 接口原型

|  |
| --- |
| public static void releaseResouce(); |

* 参数说明

无

* 使用示例

在页面销毁前释放资源

|  |
| --- |
| /\*\*  \* 在界面销毁时销毁浮标，释放账号信息，注销广播  \*/  @Override  **protected** **void** onDestroy() {  **super**.onDestroy();  // 仅当MainActivity为最后一个Activity时，才释放资源  **if** (isTaskRoot()) {  // 清空帐号资源  OpenHwID.*releaseResouce*();  }  } |

## 华为支付接口

### 订单数据生成

其中sign值的生成算法，参见商品订单签名3.2.5.1，商户需要构造一个Map (比如HashMap），写入几个支付接口必需的键/值对。**除了sign和声明不参与签名的字段都要签名，用私钥签名。**

请求参数列表

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数描述** | **是否可选** |
| userName | String(50) | 支付sdk的商户名称。不能为空 | 必填  **不参与签名** |
| userID | String | 支付ID。不能为空  在开发者联盟上**获取的支付 ID** | 必填 |
| applicationID | String | 应用ID。不能为空  在开发者联盟上获取的APP ID | 必填 |
| amount | String | 商品所要支付金额。不能为空  格式为：元.角分，最小金额为分，**保留到小数点后两位。例如：20.00**，此金额将会在支付时显示给用户确认)， | 必填 |
| productName | String(50) | 商品名称。不能为空  此名称将会在支付时显示给用户确认 注意：该字段中不能包含特殊字符，包括 # " & / ? $ ^ \*:) \ < > | , = 回车 换行 | 必填 |
| requestId | String(30) | 开发者支付订单号。**注：最长30字节。**不能为空  其值由开发者定义生成，用于标识一次支付请求**，每次请求需唯一，不可重复**。  支付平台在服务器回调接口中会原样返回requestId的值。  注意：该字段中不能包含特殊字符，包括 # " & / ? $ ^ \*:) \ < > | , = 回车 换行 | 必填 |
| productDesc | String(100) | 商户对商品的自定义描述 。不能为空  注意：该字段中不能包含特殊字符，包括 # " & / ? $ ^ \*:) \ < > | , = 回车 换行 | 必填，**不能为空** |
| extReserved | String(100) | 商户侧保留信息，输入的话在回调接口中原样返回。**注：最长100字节。** | 选填，**不参与签名** |
| sign | String | RSA签名（除了sign和声明不参与签名的字段都要签名，用私钥签名）  不能为空 | 必填 |
| screentOrient | int | 支付屏幕方向  1：竖屏（默认值）  2：横屏 | 选填，不参与签名 |
| sdkChannel | int | 应用类型  0 代表自有应用，无渠道  1 代表智汇云渠道  2 代表预装渠道  3 代表游戏吧  错误或未输入取缺省值：1 | 选填，**不参与签名** |
| environment | String | 服务器连接环境。  取值如下：  ENV\_LIVE：正式环境。  默认为正式支付环境，暂时不支持测试环境 | 选填，**不参与签名** |
| notifyUrl | String | 服务器异步通知页面路径。  华为服务器收到后**检查该应用有无在开发者联盟配置回调URL，如果配置了则使用应用配置的URL**，否则使用notifyUrl作为该次支付的回调URL **强烈建议在开发者联盟配置回调URL！** | 选填,**不参与签名** |
| serviceCatalog | String | 服务目录,直接填写:  X6 | 必填，**不参与签名** |
| urlver | String | Callback URL 版本，不输入或错误情况下，则取缺省值1：  1：缺省回调接口版本  2：回调接口版本2，回调接口中增加如下参数：  ordeTime: 下单时间  tradeTime: 交易时间  bankId: 银行编码-支付通道信息  accessMode: 接入方式  spending: 费用 | 选填，**不参与签名** |

### 调用安全支付

准备好参数后，即可调用安全支付进入支付流程并获得调用结果，关键代码如下：

|  |
| --- |
| IHuaweiPay payHelper = **new** MobileSecurePayHelper();  /\*\*  \* 开始支付  \*/  payHelper.startPay(activity, payInfo, handler); |

详情见Demo中GameActivity.java

**注：此步骤开始调用支付服务，会涉及华为安全支付服务APK的下载操作，需要开发者根据Demo中示例代码进行开发，对应参考类：**

1. **GameActivity.java：该类封装了整个调起与接收支付结果的全过程；**

### 支付结果获取和处理

调用安全支付后，将通过两种途径获得支付结果：

* 由startPay函数中的回调onFinish返回,支付结果保存在Map里面。

所有的返回参数请见如下支付结果返回参数列表。

支付结果返回参数列表

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **参数类型** | **参数说明** | **备注** |
| returnCode | String | “0”表示成功，详见3.2.4支付结果返回码 | 必填 |
| userName | String(50) | 开发者支付请求的商户昵称，原样返回 | 必填 |
| orderID | String | 支付平台订单号 | 选填 |
| requestId | String(30) | 开发者支付订单号。原样返回支付请求中requestId的值。 | 必填 |
| amount | String | 商品支付金额(格式为：元.角分) 保留到小数点后两位 | 必填 |
| errMsg | String(50) | 支付返回描述信息 | 可选 |
| time | String(15) | Java时间戳 (自1970年1月1日0时起的毫秒数) | 必填 |
| sign | String | 开发者签名，开发者需要拿公钥进行验签。 | 选填 |

* 华为服务器通知

商户需要提供一个http协议的接口，在开发者联盟上设置为回调URL，支付成功后通过该URL通知CP，详见《支付成功回调说明.pdf》。

### 支付结果返回码类表

|  |  |
| --- | --- |
| **错误代码** | **含义** |
| 0 | 成功 |
| -1 | 失败 |
| 30000 | 用户中途取消了支付 |
| 30001 | 参数或参数类型错误 |
| 30002 | **支付结果查询超时**（建议此时客户端去服务器查询订单是否支付成功） |
| 30004 | *非法请求* |
| 30005 | 网络连接异常 |
| 30006 | 系统升级 |
| 30099 | 系统错误 |

### RSA签名和验签

建议：签名和验签尽量在商户服务器端进行，同时一些敏感数据（如公私钥等）也应存储在服务器端，避免可能的安全隐患。

为了方便商户接入安全支付服务，我们将签名和验签的方法封装在IHuawiepay.jar中提供给大家使用， 具体使用方法可参考demo中支付模块。

#### 商品订单签名

(a)生成待签名字符串

集成IHuawiepay.jar后，请引用HuaweiPayUtil类的方法：

public static String getSignData(Map<String, String> params)

例子：

出售商品（productName）: MF唱片

价格（amount）:0.01(元)

订单号（requestId）: 2013-07-27-10-04-29-272

商品描述（productDesc）：【正版】MF唱片 HIFI毒药4 毒药涅磐再造 海洛 因新4号HD天碟1CD

支付ID（userID）：900086000000010204

应用ID（applicationID）：10003056

SDK来源(sdkChannel):0 (固定)

url版本(urlver)：1（固定）

集成IHuawiepay.jar后，请引用HuaweiPayUtil类的方法：

**public static** String getSignData(Map params)

params：待签名map

注：不要将空值（包括null和””情况）的参数包含到待签名数据中

(b) 对商品信息进行rsa签名

集成IHuawiepay.jar后，请引用Rsa类的方法：

**public static** String sign(String content, String privateKey)

String content：待签名字符串

String privateKey：私钥，从开放联盟开通支付权益时获得

#### 支付结果验签

使用Rsa类进行验签：

**public static boolean** doCheck(String content, String sign, String publicKey)

String content：待验签的字符串

String sign：签名值

String publicKey：公钥

返回值：验签成功则返回true，反之返回false.

以下是一个订单支付成功完整信息的示例：

{"sign":"EnyvN2+yPoHeVsoUmfVgU+pWVuiJHDVjORuq083+coXMOJRK+EthkAJdZp30IKWnlCGBFF3N97YvMzjBeqcyJA==","amount":"0.01","returnCode":"0","time":"1374890975397","orderID":"A201307271007505018046FC","requestId":"2013-07-27-10-04-29-272","errMsg":"success","userName":"华为"}

# 代码混淆

如果应用需要混淆代码，为了保证SDK的正常使用，请不要混淆SDK的jar包。