# SDK接入说明

|  |  |  |  |
| --- | --- | --- | --- |
| 文档 | 版本 | 修改 | 日期 |
| 《SDK接入说明》 | 3.0.2 | 闫世超 | 2016-07-21 |
|  | 3.0.3 |  | 2016-08-08 |

[SDK接入说明 1](#_Toc456625698)

[一、SDK文件说明 1](#_Toc456625699)

[二、 SDK使用步骤 2](#_Toc456625700)

[1、导入项目 2](#_Toc456625701)

[2、添加SDK文件 3](#_Toc456625702)

[3、修改清单文件 5](#_Toc456625703)

[4、调用SDK接口 9](#_Toc456625704)

[三、设置登录和支付回调接口 13](#_Toc456625705)

[四、测试 13](#_Toc456625706)

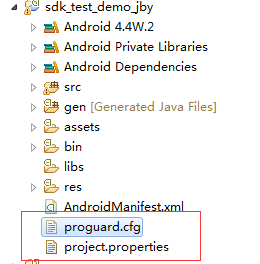
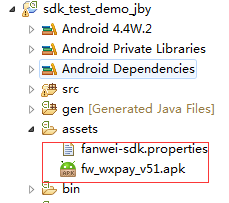
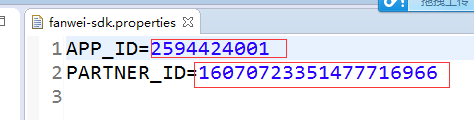
[五、推广发布 13](#_Toc456625707)

## 说明

该文档阅读对象为，接入sdk的开发人员。

该说明文档为通用sdk版本说明文档，

对接含有聚宝云支付功能的sdk时候应该注意一下几点：

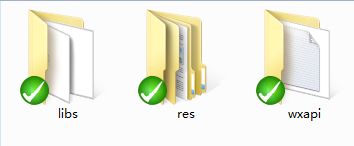
1. 该文档仅仅作为参考。
2. Sdk用到的jar和sdk用到的资源文件等，以demo为准，从演示demo拷贝。
3. 清单文件配置直接从demo里面拷贝。
4. 混淆，删除这两个文件
5. 拷贝这两个文件到项目中去
6. 
7. 修改这两个参数。

新APPID:2593966001新商户号：16061543051670158467

旧Appid： 2594424001旧商户号： 16070723351477716966。Appid和商户号联运方提供。

8.非聚宝云的微信和支付宝功能，对接时候不用测试。

## 一、SDK文件说明



Sdk包含的三个文件夹

libs目录为接入sdk需要导入的jar包

paysdk-×××.jar(基本)

libammsdk.jar(微信用)

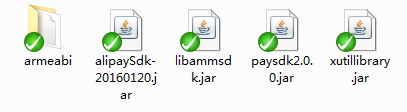
banksdk.jar（威富通）

alipaySdk-20160120.jar(支付宝用)

xutillibrary.jar(http与图片加载框架)

wxapi微信回掉专用目录。

res 目录 sdk资源文件目录



Libs目录包含的文件



Armeabi下文件

## 二、 SDK使用步骤

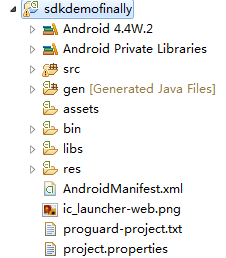
以下以Eclipse为开发IDE进行范例说明：

准备：需要安装jdk和eclipse下配置安卓sdk;

没有安装的需要网上查找相关教程。

### 1、导入项目

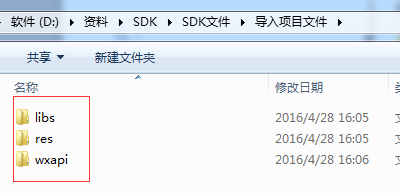
开发者将项需要集成sdk的项目导入eclipse。



### 2、添加SDK文件

将SDK文件添加到项目指定目录

sdk文件夹目录



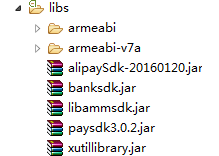
#### （1.）添加libs和res目录

复制libs，res两个文件夹，粘贴到项目根目录覆盖。

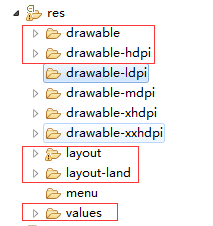
覆盖之后，项目原有libs目录和res目录对应多了sdk的libs目录和res目录中的内容。

为了避免重复，sdk资源文件命名多以mch\_前缀，并且命名偏长。

lib文件下的所有文件如下：



res下的所有文件如下：



将sdk的libs目录和res目录下的所有文件都拷贝到项目的libs目录和res目录下对应的目录中。

#### （2.）导入jar包

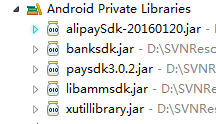
一般情况下，jar包拷贝到libs目录会自动导入jar包；

如果没有自动导入的话，请打开libs目录；

选中paysdk.jar，libammsdk.jar，alipaySdk-20160120.jar；

右击 -> build path -> add to build path；导入成功。

成功导入



#### （3.）添加wxapi目录

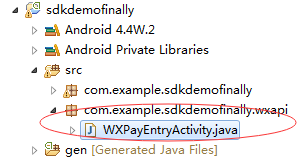
打开src目录下程序主包名所在目录；

拷贝wxapi文件夹到该目录；

打开WXPayEntryActivity.java，修正WXPayEntryActivity.java所在的包；

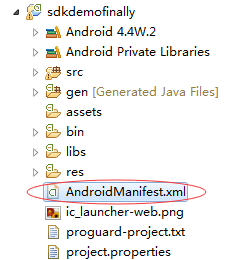
WXPayEntryActivity不报错；wxapi目录添加成功。

添加wxapi之后



### 3、修改清单文件

文件位置



#### （1.）添加权限

打开项目AndroidManifest.xml文件，在mainfest内application外添加以下代码：

<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.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />

#### （2.）添加activity声明：

复制以下代码到application节点里面。

<activity

android:name=*"com.mchsdk.paysdk.TestActivity"*

android:screenOrientation=*"landscape"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.ChoosePayActivity"*

android:launchMode=*"singleTop"* android:theme=*"@android:style/Theme.NoTitleBar.Fullscreen"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.PTBPayResultActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCPacksActivity"*

android:launchMode=*"singleInstance"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCPersonalInfoActivity"*

android:launchMode=*"singleTask"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCChangePasswordActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCMoneyRecordActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCAddPtbMoneyActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCGameRecordActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.mchsdk.paysdk.activity.MCBindPhoneActivity"*

android:launchMode=*"singleTop"* >

</activity>

<activity

android:name=*"com.alipay.sdk.app.H5PayActivity"* android:configChanges=*"orientation|keyboardHidden|navigation"*

android:exported=*"false"*

android:screenOrientation=*"behind"* android:windowSoftInputMode=*"adjustResize|stateHidden"* >

</activity>

<activity

android:name=*"com.alipay.sdk.auth.AuthActivity"* android:configChanges=*"orientation|keyboardHidden|navigation"*

android:exported=*"false"*

android:screenOrientation=*"behind"* >

</activity>

<service android:name=*"com.mchsdk.paysdk.server.MCHFloatService"* >

</service>

<activity android:name=*"com.mchsdk.paysdk.activity.MCForgetPasswordActivity"* >

</activity>

<activity

android:name=*"com.example.sdk\_test\_demo.wxapi.WXPayEntryActivity"*

android:exported=*"true"*

android:launchMode=*"singleTop"*

android:screenOrientation=*"portrait"*

android:theme=*"@android:style/Theme.Translucent"* >

</activity>

<!-- *paysdk\_signkey为加密之后的sdk客户端和sdk服务器验签用的key。*

*这个key可以更换，如过不更换就用C*gsPEgsLEg8LCg，这里的值不能全为数字 -->

<!-- CgsPEgsLEg8LCg这个paysdk\_signkey要和服务器一致， 生成的密码是 gnauhcgnem -->

<meta-data

android:name=*"paysdk\_signkey"*

android:value=*"CgsPEgsLEg8LCg=="* />

<!-- 这里面填写平台服务器的IP地址 -->

<meta-data

android:name=*"paysdk\_ipaddress"*

android:value=*"这里替换为你的sdk服务器地址/sdk.php"* />

#### （3.）修改paysdk\_ipaddress节点

按照提示将服务器域名填写在该节点的value属性下。

### 4、调用SDK接口

调试环境

MCLog.*isDebug* = **true**;

生产环境

MCLog.*isDebug* = **false**;

#### (1) 登录功能使用

import com.mchsdk.paysdk.MCApiFactory;

import com.mchsdk.paysdk.IGPUserObsv;

/\*\* \* 登录回调接口 \*/

private IGPUserObsv mUserObsv = new IGPUserObsv() {

@Override

public void onFinish(GPUserResult result) {

switch (result.getmErrCode()) {

case GPUserResult.USER\_RESULT\_LOGIN\_FAIL:

writeLog("登录回调:登录失败");

break;

case GPUserResult.USER\_RESULT\_LOGIN\_SUCC:

MCApiFactory.getMCApi().startFloating(MainActivity.this);

String key = "mengchuang";

// 登录用户id

String accountNo = result.getAccountNo();

// 登录签名

String sign = result.getSign();

// 登录时间戳

String timeStamp = result.getTimeStamp();

// 登录用户账号

String account = result.getAccount();

String tempString = accountNo.toString() + account.toString()

+ key + timeStamp.toString();

String temp2 = PaykeyUtil.stringToMD5(tempString);

writeLog("登录回调:登录成功sign = " + sign);

writeLog("登录回调:登录成功temp = " + temp2);

break; }

}

};

/\*\* \* 接口调用 \*/

MCApiFactory.getMCApi().startlogin(context,mUserObsv);

#### (2)支付调用的方法：

import com.mchsdk.paysdk.MCApiFactory;

import com.mchsdk.paysdk.entity.OrderInfo;

import com.mchsdk.paysdk.payCallback;

// 支付结果回调

public payCallback payCallback = new payCallback() {

public void callback(int result) {

if (result == 1) {

//支付成功

return;

}

}

};

// 支付接口调用

OrderInfo o = new OrderInfo();

//物品价格,单位分。

o.setAmount(1);

//cp用于确认交易信息，给玩家发送装备，

//这个参数会在支付成功之后透传给游戏方

o.setExtendInfo("");

//物品描述

o.setProductDesc("");

//物品名称

o.setProductName("12");

MCApiFactory.getMCApi().pay(MainActivity.this, getFragmentManager(), o, payCallback);

#### (3)悬浮窗

显示悬浮窗需要在对应的Activity添加如下内容：

MCApiFactory.getMCApi().startFloating(MainActivity.this);

MCApiFactory.getMCApi().stopFloating(MainActivity.this);

#### (4)退出接口回调：

普通退出回调接口：

**private** IGPExitObsv mExitObsv = **new** IGPExitObsv() {

@Override

**public** **void** onExitFinish(GPExitResult exitResult) {

**switch** (exitResult.mResultCode) {

**case** GPExitResult.***GPSDKExitResultCodeError*** :

*writeLog*("退出回调:调用退出弹框失败");

**break**;

**case** GPExitResult.***GPSDKExitResultCodeExitGame*** :

// 在这里填写退出操作

// System.exit(0);

// finish();

**break**;

**case** GPExitResult.***GPSDKExitResultCodeCloseWindow*** :

*writeLog*("退出回调:调用关闭退出弹框");

**break**;

}

}

};

退出方法调用：

MCApiFactory.*getMCApi*().exit(MainActivity.**this**, mExitObsv);

#### (5)退出接口回调：

Sdk悬浮穿里面的个人中心退出登录回调接口:

IGPExitObsv exitFromPersonInfo = **new** IGPExitObsv() {

@Override

**public** **void** onExitFinish(GPExitResult gpExitResult) {

/\*\* 个人中心里面里面点击退出按钮 \*\*/

**if** (GPExitResult.***GPSDKExitResultCodeExitGame***

== gpExitResult.mResultCode) {

MCLog.*i*("MainActivity", "执行个人中心里面退出登录方法回调");

ToastUtil.*showToast*(getApplicationContext(), "执行个人中心里面退出登录方法回调");

}

}

};

接口调用：

1.初始化个人中心里面点击退出登录之后执行的退出回调的接口

MCApiFactory.*getMCApi*()

.initExitFromPersonInfoParams(exitFromPersonInfo);

2.如果想要点击个人中心里面的退出登录之后跳转到指定的activity，那么就在这个activity里面初始化这个方法

MCApiFactory.*getMCApi*()

.initExitFromPersonInfoParams(MainActivity.**class**);

3.如果两个初始化方法都不执行，点击个人中心里的退出登录，程序返回到桌面

#### (6)初始化设置默认渠道参数：

这个方法用于设置打包的参数，这些参数来自平台后台，在apk上传到平台打包之后产生。在这里设置是为了省去平台打包步骤，方便测试。

设置以后，如果apk没有在平台打包，sdk会调用这里面设置的参数。

MCApiFactory.*getMCApi*().initTestParams("0"// 渠道id

, "自然注册"// 渠道名称

, "12"// 游戏id

, "决战中洲"// 游戏名称

, "520D9380CF1B71E89");// 游戏appid

/\*\* 渠道信息优先级，真实打包 > 这里填写的测试信息 > sdk里面的测试参数

#### (7)调用执行sdk注销登录的方法：

MCApiFactory.getMCApi().sdkLogOff();

#### (8)退出登录弹窗：

/\*\* 退出登录弹窗 \*/

MCApiFactory.getMCApi().dismisDialog();

该方法用于解决，手机待机黑屏之后，屏幕再次亮起的时候，登录弹窗点击没反应的问题。

开发者监听屏幕状态，再次亮起时候，退出登录弹窗重新调用登录方法。

## 三、设置登录和支付回调接口

客户端集成好sdk之后，游戏方按照《游戏通知接口文档》向联运提供登录回调接口和支付回调接口，联运后台的 - 游戏 - 设置 里面将这两个接口设置上传即可。

## 四、测试

以上步骤完成之后，将集成sdk对应的apk上传联运平台打渠道包，打包之后的app包含渠道对应标识。该标识用于渠道推广线上统计等。模拟玩家通过推广链接下载改游戏。运行游戏客户端支付功能，游戏支付回调接口能够接收支付成功的信息。至此，sdk支功能集成对接完成。

## 五、推广发布

按照以上流程可以上传多款游戏，可以分发多个推广员推广，玩家通过推广链接下载游戏。实现后台游戏登录充值分成统计管理。