

第三方支付接入文档（客户端—android 版）

前言

编写目的：

Android 游戏 SDK 使用说明及其 SDK，是为了简化合作伙伴对游戏计费等功能开发流程。

适用对象及范围：

本文档主要面向需要接入指易付第三方支付服务（客户端）的商户的开发人员。

运行环境：

Android SDK: SDK1.6 及以上

IDE: Eclipse with ADT

目录

第一章 指易付业务流程介绍.....	3
第二章 接入准备	4
2.1 工程依赖 && APK 打包	4
2.1.1 方式一：手动配置	4
2.1.2 方式二：工具配置	7
第三章 接口说明	9
3.1 客户端接口概要	错误！未定义书签。
3.2 SKYPAYSERVER 实例化接口：	9
3.3 SKYPAYSERVER 初始化接口（传入回调）	9
3.4 启动接口	9
3.5 请求付费接口	9
3.6 支付结果回调	10
3.7 调用示例	10
第四章 第三方参数说明.....	10
4.1 订单格式	10
4.2 第三方订单参数、	11
4.3 第三方支付结果参数 (MSG. WHAT=1000)	12
第五章 常量名称对应表.....	13
第六章 注意事项	14
6.1 定价规则	14
6.2 商户号不存在	14
6.3 参数确认	14
6.4 关于 .SO	14
6.5 签名有误	15

第一章 指易付业务流程介绍

支付过程中所涉及到的对象及他们之间的关系：

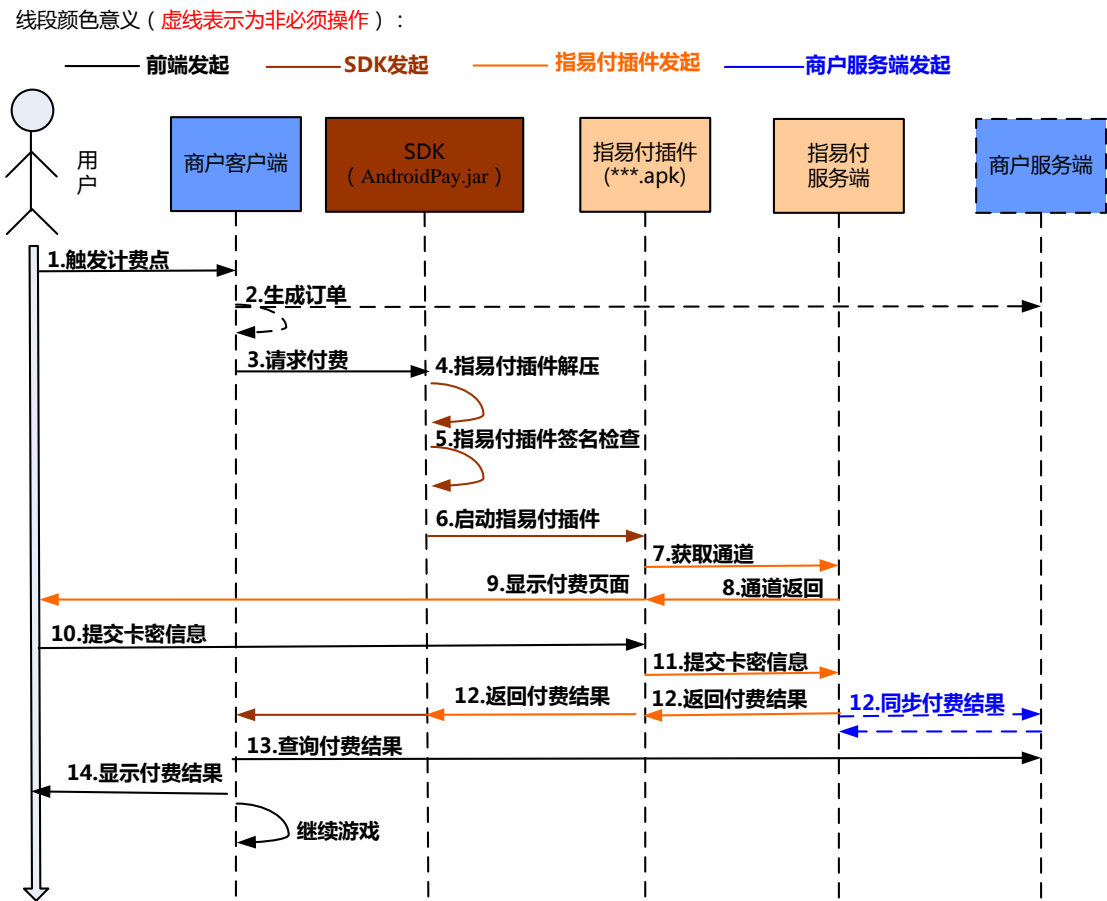


图 1-1 指易付支付流程图

第二章 接入准备

2.1 工程依赖 && APK 打包

2.1.1 方式一：手动配置

步骤一 将相关文件放到 assets 目录

如图：

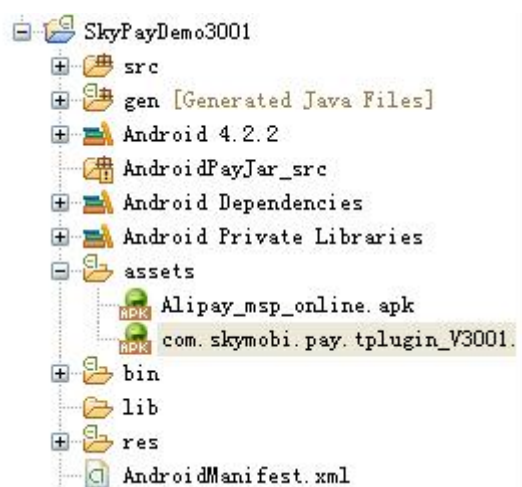


图 2-1 将 apk 放到 assets 下示意图

包括

1、com.skymobi.pay.ipugin_V***.apk

支付插件，必须放在 assets 目录下。

2、Alipay_msp_online.apk

支付宝插件，如果没打包该插件，请求支付宝付费时将使用支付宝 wap 页面支付，建议打包到 assets 目录，以提高用户体验。

步骤二 工程依赖

对 OptimizationSKAndroidPayRes 工程进行工程依赖。（若添加失败，请检查是否导入依赖工程，且依赖工程与游戏项目要在同一目录下）

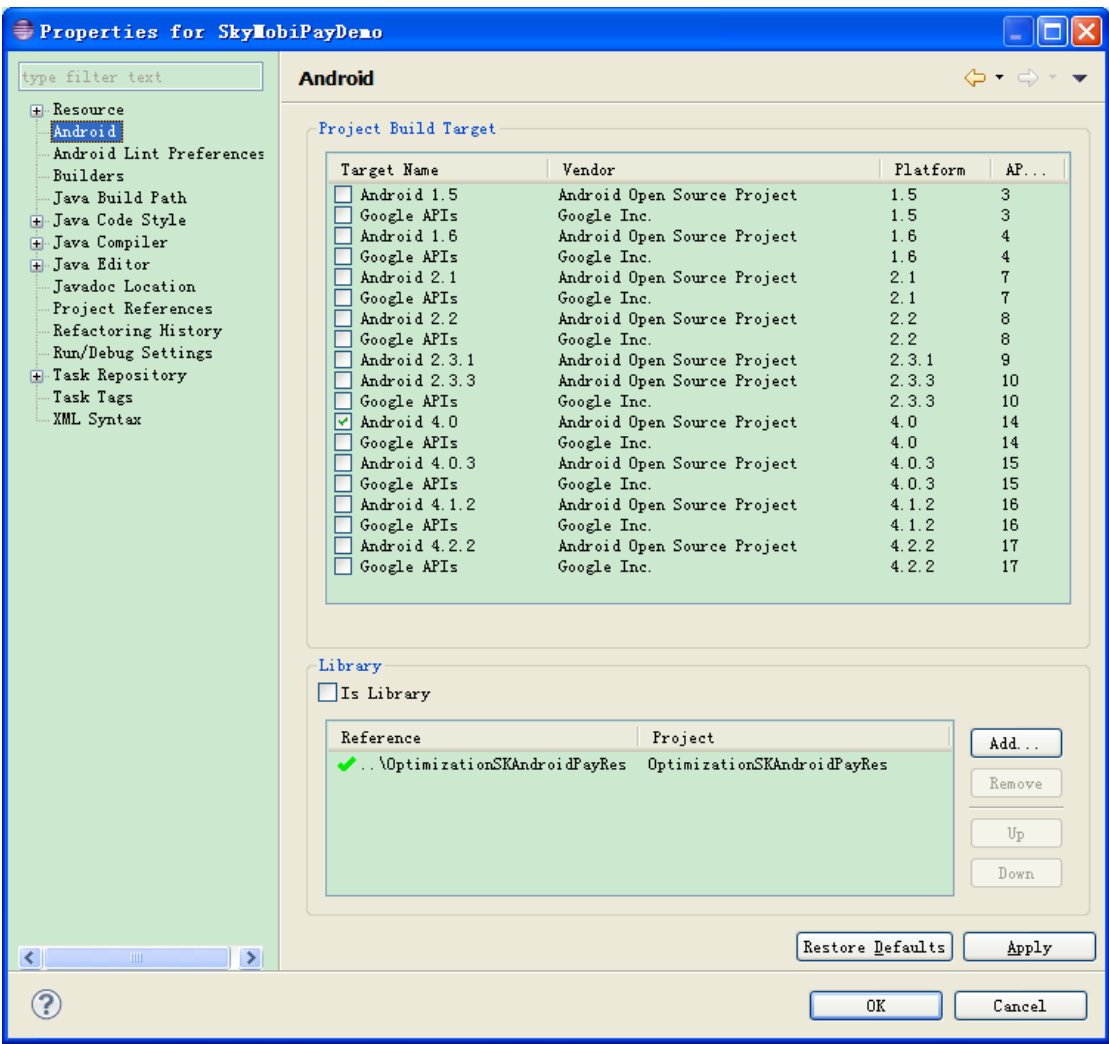


图 2-2 工程依赖示意图

步骤三 添加 AndroidPay.jar

将 AndroidPay.jar 包添加到工程 Java Build Path 的 libraries 中

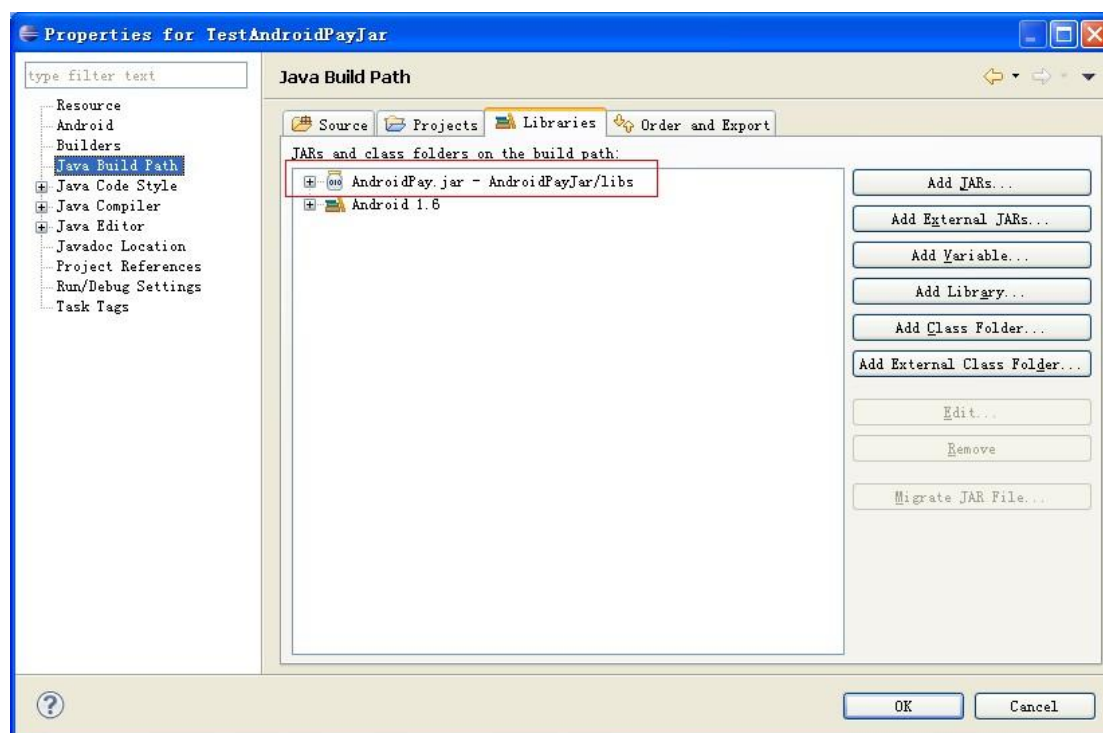


图 2-3 添加 AndroidPay.jar 示意图

步骤四 配置 AndroidManifest.xml

将 OptimizationSKAndroidPayRes 工程中 AndroidManifest.xml 的相关配置，复制到您的工程目录下的相同文件中（含权限配置）。

若已经继承其他 application 类，则可以在应用的 application 类中，进行调用，需继承 com.skymobi.pay.app.PayApplication 这个类，在应用的 AndroidManifest.xml 中的 application 节点配置信息，如图 2-4 所示。注意一定要在 application 中 onCreate 函数中去调用 applicationOnCreat 接口，如图 2-5 所示。



图 2-4 在 application 结点配置信息

```

5 package com.skymobi.pay.demo;
6
7 import com.skymobi.pay.app.PayApplication;
8
9 import android.app.Application;
10
11 /**
12  * @author Leo.wu
13  *
14  * 2013-5-8
15  */
16 public class SkyMobiPayDemoApplication extends Application {
17     private PayApplication payApplication = new PayApplication();
18     @Override
19     public void onCreate() {
20         super.onCreate();
21         payApplication.applicationOnCreat(getApplicationContext());
22     }
23 }

```

图 2-5 在 oncreat 函数中调用 applicationOnCreat 接口

2.1.2 方式二：工具配置

步骤一 将工具包放在工程目录下

首先将配置工具包（pay-conf-jar）放置在您的工程目录下：

将 OptimizationSKAndroidPayRes 工程中 AndroidManifest.xml 放到文件夹 pay-conf-jar 中

assets	文件夹	2013-5-17 11:18
bin	文件夹	2013-4-27 12:56
gen	文件夹	2013-4-27 13:50
libs	文件夹	2013-4-7 13:18
pay-conf-jar	文件夹	2013-3-29 13:36
proguard	文件夹	2013-4-25 14:06
res	文件夹	2013-4-25 15:36
src	文件夹	2013-3-29 13:46
.classpath	1 KB CLASSPATH 文件	2013-4-27 13:52
.project	2 KB PROJECT 文件	2013-4-25 16:21
AndroidManifest.xml	7 KB XML 文档	2013-5-15 16:45
default.properties	1 KB PROPERTIES 文件	2013-4-27 13:48
ic_launcher-web.png	51 KB PNG 图像	2013-3-29 13:28
proguard.cfg	2 KB CFG 文件	2013-3-29 13:43
proguard-project.txt	1 KB Notepad++ Document	2013-3-29 13:28

图 2-6 复制 pay-conf-jar 工具包

步骤二 编辑并运行 start.bat 文件

- 1、打开 start.bat 文件

名称	大小	类型
AndroidManifest.xml	6 KB	XML 文档
android-pay-conf-1.0.1.jar	18 KB	Executable Jar File
dom4j-1.6.1.jar	307 KB	Executable Jar File
jaxen-1.1-beta-12.jar	224 KB	Executable Jar File
pay.classpath	1 KB	CLASSPATH 文件
proguard.cfg	1 KB	CFG 文件
start.bat	1 KB	MS-DOS 批处理文件

图 2-6 start.bat 文件

2、编辑 start.bat 文件

将 start.bat 文件中的 currentProjectPath 改为您的工程所在目录的路径（绝对路径）。



图 2-7 修改 currentProjectPath

3、保存并运行 start.bat 文件，若配置成功，则运行结果如图 2-8 所示。

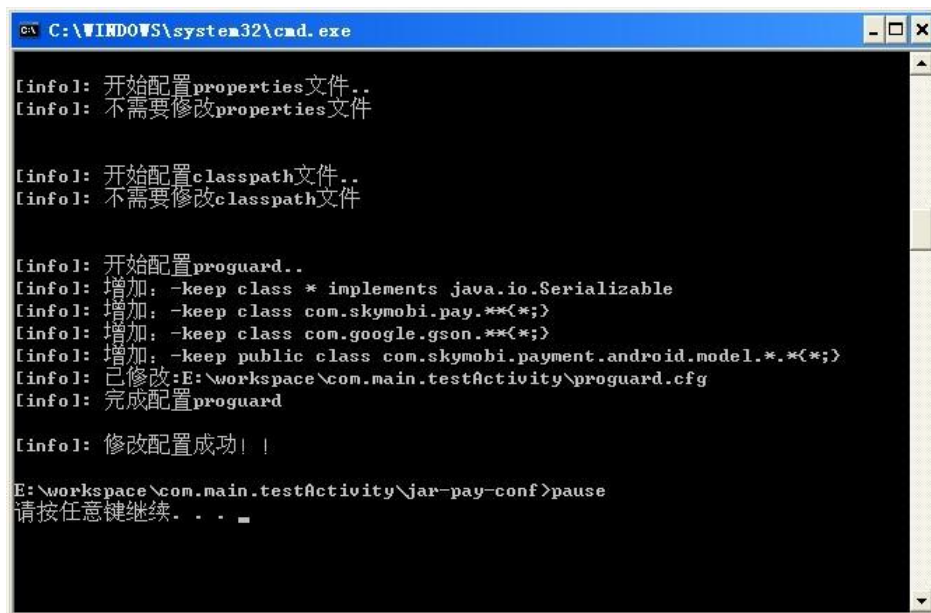


图 2-8 配置成功示意图

第三章 接口说明

AndroidPay.jar 的 SkyPayServer 实现了实例化 SkyPayServer、初始化 SkyPayServer、app 启动接口，启动计费接口，合作伙伴只需要监听计费成功或失败的状态即可。

类名：SkyPayServer

常量说明见[第五章](#)

3.1 SkyPayServer 实例化接口：

```
SkyPayServer getInstance();
```

3.2 SkyPayServer 初始化接口（传入回调）

```
int init(Handler cb);
```

返回值：

SkyPayServer. *PAY_RETURN_FAILED* //传入参数为空

SkyPayServer. *PAY_RETURN_SUCCESS* //初始化成功

支付结果将通过 Handler 回调，见[3.6 支付结果回调](#)

3.3 启动接口

```
int startUp (Activity activity, String startUpInfo);
```

参数说明：

activity	启动付费界面的activity
orderInfo	启动信息，为文本键值对格式：“payMethod=3rd&appId=*****”

此接口主要进行计费系统初始化的动作，确保在游戏启动后并有网络时立即进行调用！只调用一次。

返回值

SkyPayServer. *PAY_RETURN_SUCCESS* //初始化成功

SkyPayServer. *PAY_RETURN_PAYING_ERROR* //初始化失败，服务处于付费状态

SkyPayServer. *PAY_RETURN_FAILED* // 初始化失败，传入参数为空

3.4 请求付费接口

```
int startActivityAndPay(Activity activity, String orderInfo);
```

参数说明：

activity	启动付费界面的activity
orderInfo	订单信息，为文本键值对格式如： “ order_id=100014&merchant_id=m900&price=200&merchant_si gn =%¥1214%12 ” 订单具体参数定义见 4.1 订单格式

返回值：

SkyPayServer. *PAY_RETURN_UNINIT_ERROR* //SkyPayServer 未初始化

SkyPayServer. *PAY_RETURN_FAILED* //传入参数为空

SkyPayServer. *PAY_RETURN_SUCCESS* //接口调用成功

3.5 支付结果回调

```
mPayHandler = new PayHandler();
private class PayHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        String strRet = (String) msg.obj;
        if (msg.what == SkyPayServer.MSG_WHAT_TO_APP) {
            //支付结果，结果放在strRet里，参数存放格式为键值对
            //具体参数见4.3 第三方支付结果参数
        }
    }
}
```

3.6 调用示例

详见SkyMobiPayDemo

第四章 第三方参数说明

4.1 订单格式

文本键值对格式：对键、值大小写敏感；

格式	说明	取值示例
----	----	------

文本键值对	以 “&” 间隔开的 key=value 形式的键值对组	order_id=100014& merchant_id=m900& price=200& merchant_sign=%¥1214%12
-------	------------------------------	--

4.2 第三方订单参数

订单参数需要在服务端生成。参考 SkyPayDemo

参数名称	必填	说明	类型（长度）	备注
payMethod	是	付费方式	String	3rd: 第三方
merchantId	是	商户号	String (50)	sky 分配
merchantSign	是	商户签名	String (50)	按要求用商户密钥，对订单生成的签名
appId	是	应用编号	Int	由斯凯分配的每个应用的唯一标识号
appName	是	应用名称	String (30)	与安装后的应用名称相同
price	是	付费价格	Int	以分为单位
orderId	是	订单号	String (50)	保证唯一 ，以便日后问题查找和数据统计
systemId	是	系统号	Int	system=300021 冒泡堂 system=300024 支付中心 例：通过冒泡市场接入支付 sdk, system=300021 不确定的，可咨询斯凯技术支持
payType	是	支付类型	Int	0=注册 1=道具 2=积分 3=充值
notifyAddress	是	应用服务端通知地址	String (512)	Http 协议
appVersion	是	应用版本号	int	需要跟应用 apk 的 android:versionCode 保持一致
gameType	否	游戏类型	Int	0: 单机 1: 联网 2: 弱联网
productName	否	商品名称	String (50)	用户要支付的道具或者虚拟货币名称，如：金豆
channelId	是	渠道号	String (30)	若应用在斯凯渠道推广，则选择填写下列预定义值 冒泡市场： 1_zhiyifu_ 冒泡游戏： 9_zhiyifu_ 如果在斯凯以外的渠道推广，以 daiji_ 打头，后面

				自定义（自定义内容不能含有‘zhiyifu’）
reserved1	否	保留字段 1	String (50)	留给 cp 存储自己的变量，通知 cp 服务端的时候，会把该变量保持不变的传回去。 如此字段中包含“=”、“ ”、“/”等转义字符时，需要进行 URLEncode 操作
reserved2	否	保留字段 2	String (50)	同 reserved1
reserved3	否	保留字段 3	String (50)	同 reserved1

4.3 第三方支付结果参数(msg.what=1000)

第三方支付结果回调次数仅一次，格式如下：

支付结果	msg_code=100&3rdpay_status=***&pay_price=***&skyChargeId=***
错误	msg_code=101&error_code=***

注：付费完成，不代表付费成功，实际付费价格以服务器通知为准。

支付结果参数：

参数名称	说明	备注
msg_code	消息类型	100：支付结果 101：错误
error_code (msg_code=101)	错误码	200： 没有网络 201： 上次付费尚未完成（远程支付服务未空闲） 202： apk 下载更新失败 203： 内存不足（下载 apk 需要的内存） 204： 网络超时（检查超时） 205： 异常退出 206： 用户取消安装插件 207： 用户取消更新插件 208： 没有收到插件回调前，调用了 ondestroy(true) 209： 请求的支付方式(payMethod)异常 300： 异常退出 301： 订单参数有误（文本键值对格式不合法） 302： 没有网络 303： 没有 sim 卡 304： 内存不足（小于 10K） 305： 上次付费尚未完成（支付控制服务未空闲） 401： 订单数据有误（缺少必要字段） 402： 网络超时（获取通道）

		403: 通道数据有误 404: 软件拦截（服务端配置）
3rdpay_status (msg_code=100)	付费状态	200: 用户点击取消，未进入充值提交步骤 201: 用户提交订单成功 202: 用户提交订单失败 203: 用户真正销卡支付成功 204: 用户真正销卡支付失败
pay_price	付费结果	指易付插件返回给 cp 客户端的付费结果。 pay_price≠0 付费成功，最终的计费价格以服务端通知为主
skyChargeId	支付流水号	

第五章 常量名称对应表

调用付费相关接口，同步返回值

名称	类型	值	含义
<i>PAY_RETURN_SUCCESS</i>	Int	0	调用成功
<i>PAY_RETURN_FAILED</i>	Int	-1	调用失败（一般因为传入的某些参数为null）
<i>PAY_RETURN_PAYING_ERROR</i>	Int	1	正在付费
<i>PAY_RETURN_UNINIT_ERROR</i>	Int	2	没有初始化（没调用过 <code>init(Handler cb)</code> ）

通过 handle 回调返回的 msg.what

名称	类型	值	含义
<i>MSG_WHAT_TO_APP</i>	Int	1000	

返回结果（key-value 格式）msg.obj 里包含的参数名称

名称	类型	值	含义
<i>PARAM_NAME_MSG_CODE</i>	String	"msg_code"	消息类型 100: 付费结果， 101: 错误码
<i>PARAM_NAME_ERROR_CODE</i>	String	"error_code"	错误码
<i>PARAM_NAME_PAY_PRICE</i>	String	"pay_price"	实际付费价格
<i>PARAM_NAME_3RDPAY_STATUS</i>	String	"3rdpay_status"	第三方付费状态

消息类型定义（msg_code 取值）

名称	类型	值	含义
<i>MSG_CODE_PAY_RESULT</i>	Int	100	返回付费结果
<i>MSG_CODE_ERROR</i>	Int	101	返回错误

第三方返回的付费状态（3rdpay_status 取值）

名称	类型	值	含义
<i>THIRD_PAY_STATUS_DO_NOTHING</i>	Int	200	用户点击取消，未进入充值提交步骤
<i>THIRD_PAY_STATUS_SUBMIT_SUCCESS</i>	Int	201	用户提交订单成功
<i>THIRD_PAY_STATUS_SUBMIT_FAILED</i>	Int	202	用户提交订单失败
<i>THIRD_PAY_STATUS_PAY_SUCCESS</i>	Int	203	用户真正销卡支付成功
<i>THIRD_PAY_STATUS_PAY_FAILED</i>	Int	204	用户真正销卡支付失败

第六章 注意事项

6.1 定价规则

第三方支付支持的最小价格都是 1 元。

6.2 商户号不存在

现网环境的商户号和商户密钥需要重新向 sky 申请。

6.3 参数确认

APPID、SYSTEMID、CHANNELID 请与 sky 确认

6.4 关于 .so

依赖工程中的 .so 文件会打包到游戏的 apk 当中,如果游戏需要 armeabi-v7a 目录,请在依赖工程 (OptimizationSKAndroidPayRes) 中的 libs 目录中复制粘贴一个 armeabi 目录改名为 armeabi-v7a。如图 2-9 所示。

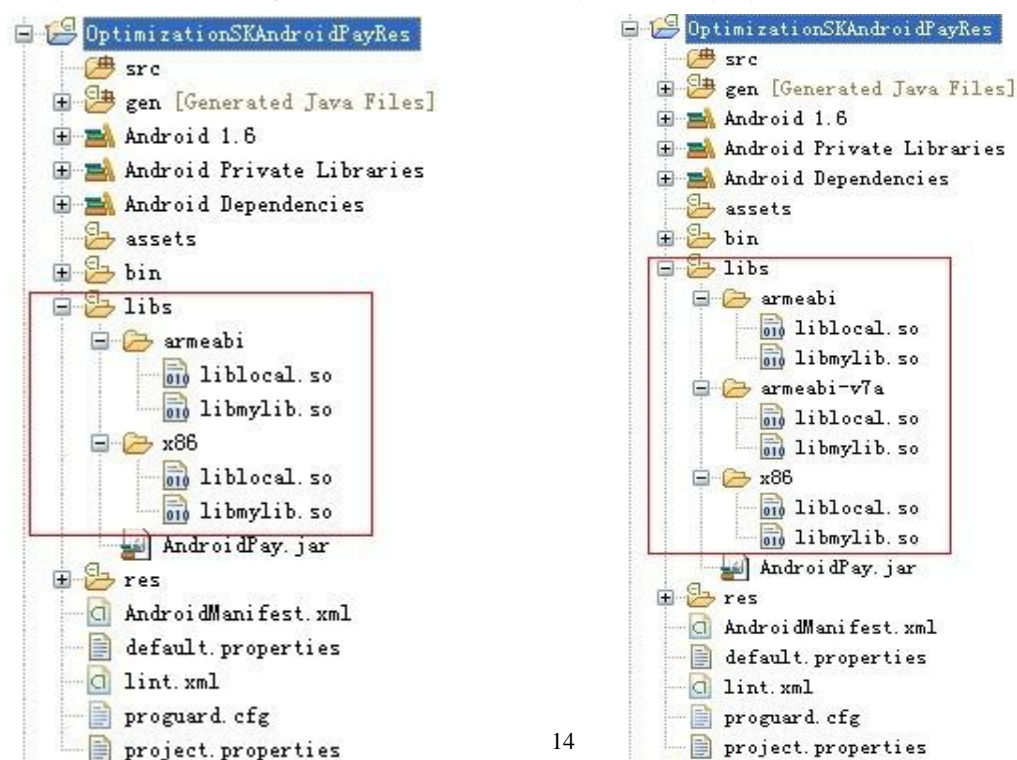


图 2-9 左图（修改前）右图（修改后）

6.5 签名有误

- （1）检查商户号和商户密钥的正确性；
- （2）检查参与签名参数的顺序；
- （3）检查参与签名参数和订单里的参数是否一致；
- （4）保证以上 3 点都没有问题后，把生成的签名字符串，发给我们进行比对，查找最终原因。由于签名是经常容易出错的地方，建议把发出数据或者受到数据，签名字符都打印起来，以便快速定位问题。