

## UPAY（优贝）短信支付完整版接入说明文档



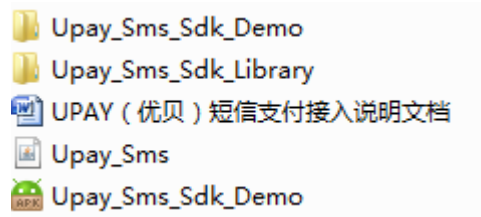
作者	版本	日期
郑翔宇	1.0.3	2013-08
李鑫宇	1.0.4	2013-08
李鑫宇	1.0.5	2013-09
李鑫宇	1.0.6	2013-09
李鑫宇	1.0.7	2013-09
李鑫宇	1.0.8	2013-11



## 目 录

一. 解压缩开发包.....	1
二. 配置 Eclipse 环境 .....	1
三. 在 AndroidManifest 中加入 SDK 必须的声明 .....	6
四. 调用短信支付界面 .....	8
五. apk 更新接口（非必选） .....	11
六. 混淆的配置 .....	12
七. 回传交易结果至开发商 BillingCallbackURI（非必选） .....	12
八. 对账与防刷接口（非必选） .....	13
九. 联系方式 .....	14

## 一. 解压缩开发包



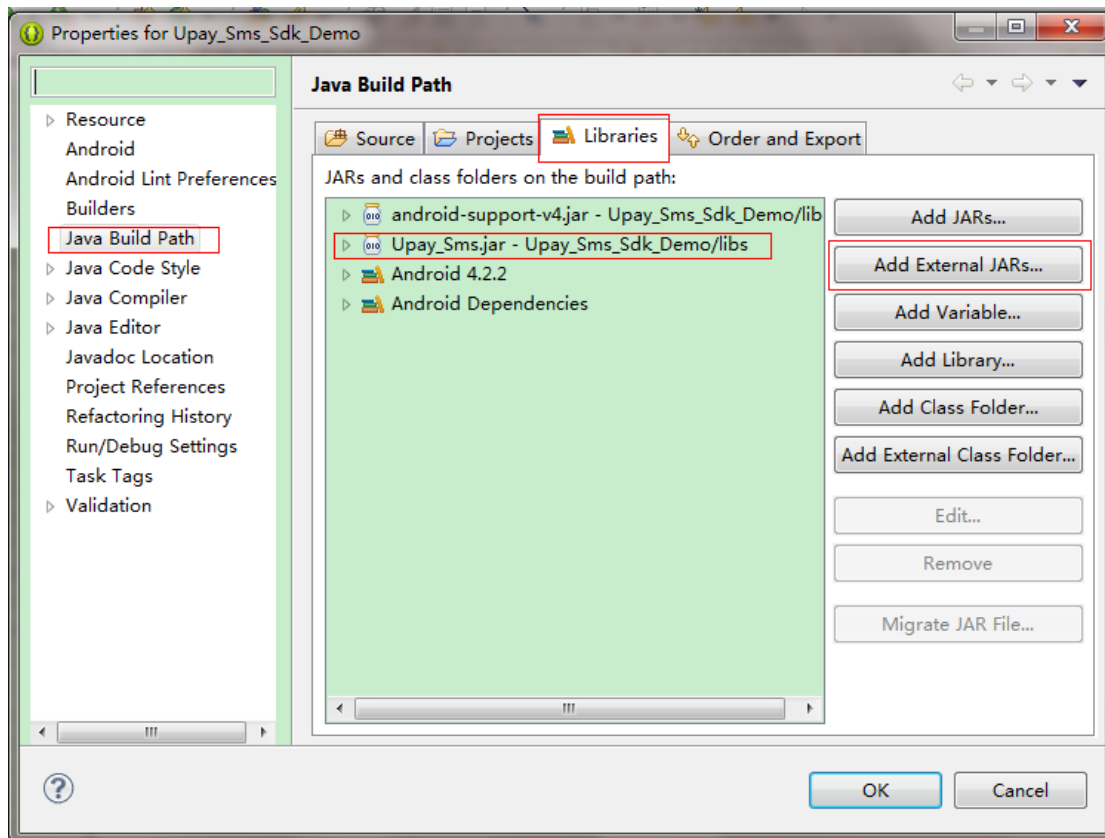
解压缩后可以看到 Upay\_Sms\_Sdk\_Demo 和 Upay\_Sms\_Sdk\_Library 文件夹，Upay\_Sms.jar 开发包，Upay\_Sms\_Sdk\_Demo.apk 示例程序。

Upay\_Sms\_Sdk\_Demo 是示例工程源码。

Upay\_Sms\_Sdk\_Library 短信支付库文件。

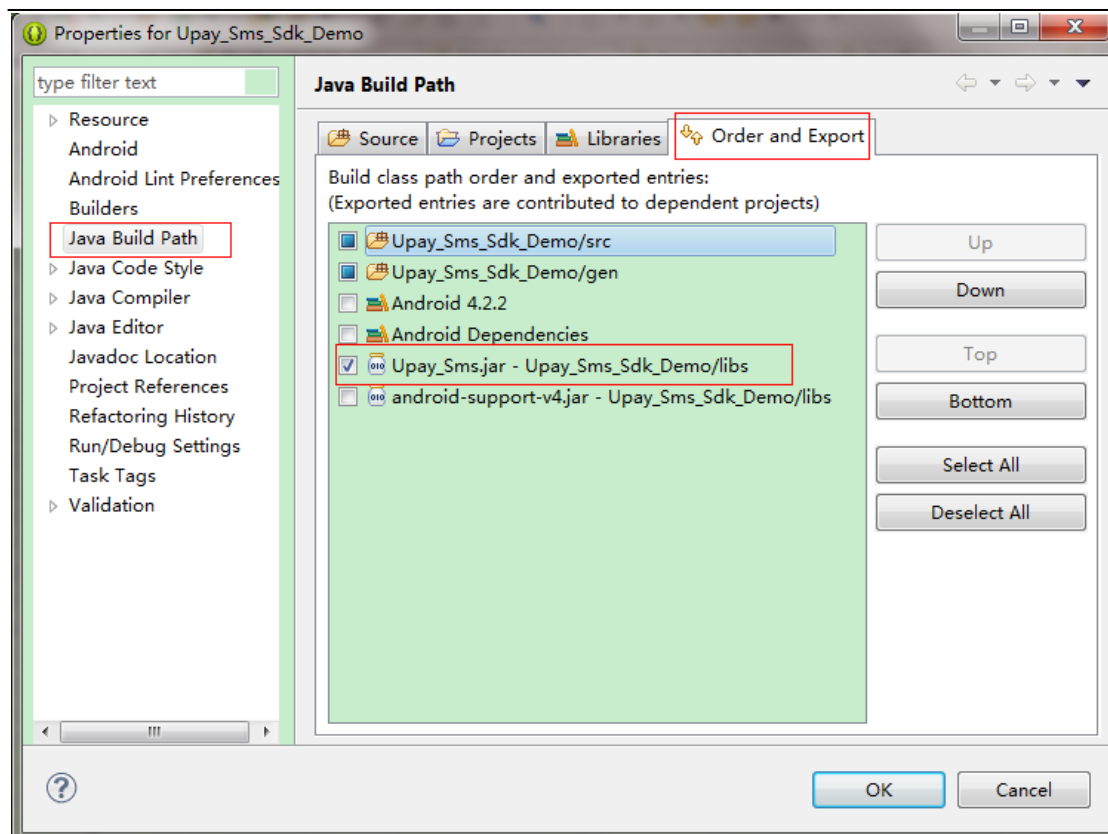
## 二. 配置 Eclipse 环境

- 1、 将 Upay\_Sms.jar 开发包复制到项目 libs 文件夹中
- 2、 右键点击项目名称
- 3、 打开 Properties 窗口
- 4、 选择 Java Build Path 选项
- 5、 选择 Libraries 标签栏
- 6、 点击 Add External JARs
- 7、 选择 libs 文件夹中的 Upay\_Sms.jar 开发包

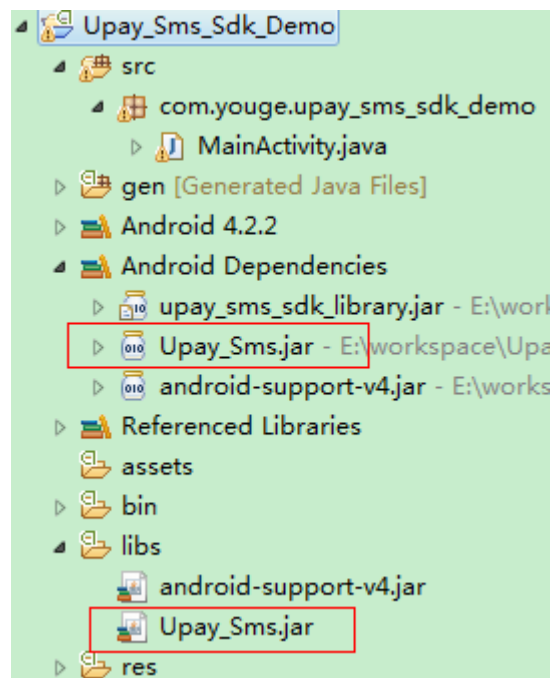


8、选择 Order and Export 标签栏

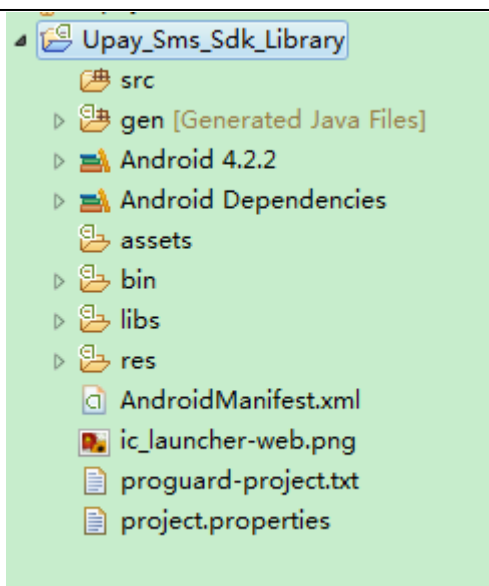
9、选中已添加的 Upay\_Sms.jar



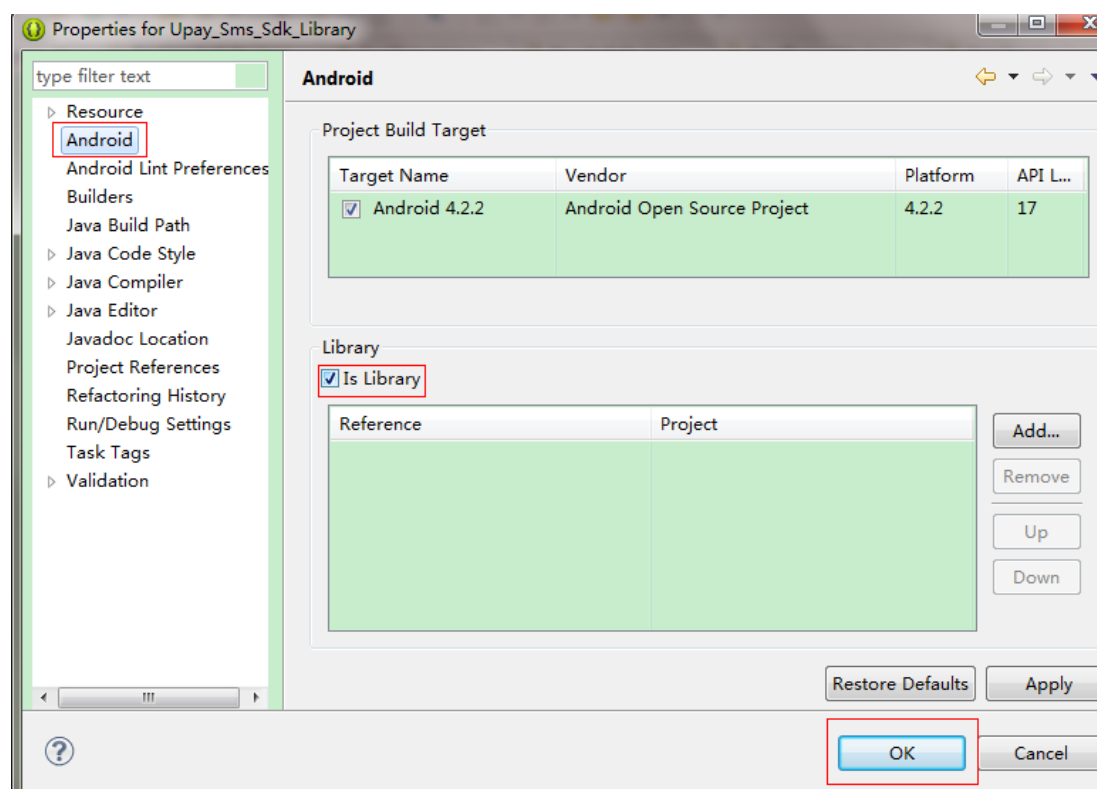
10、 点击 OK 完成嵌入



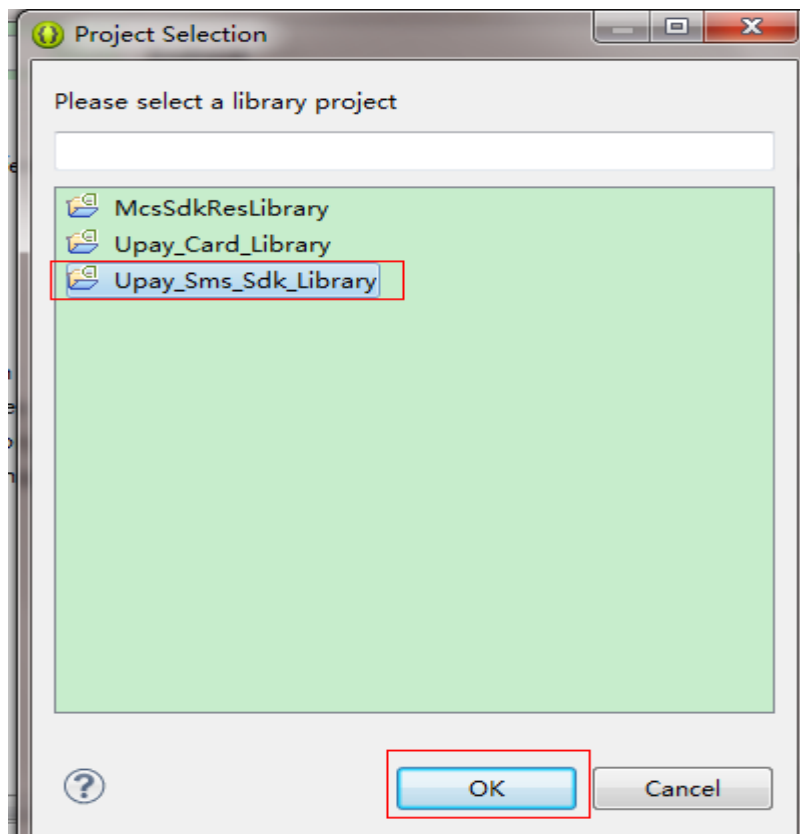
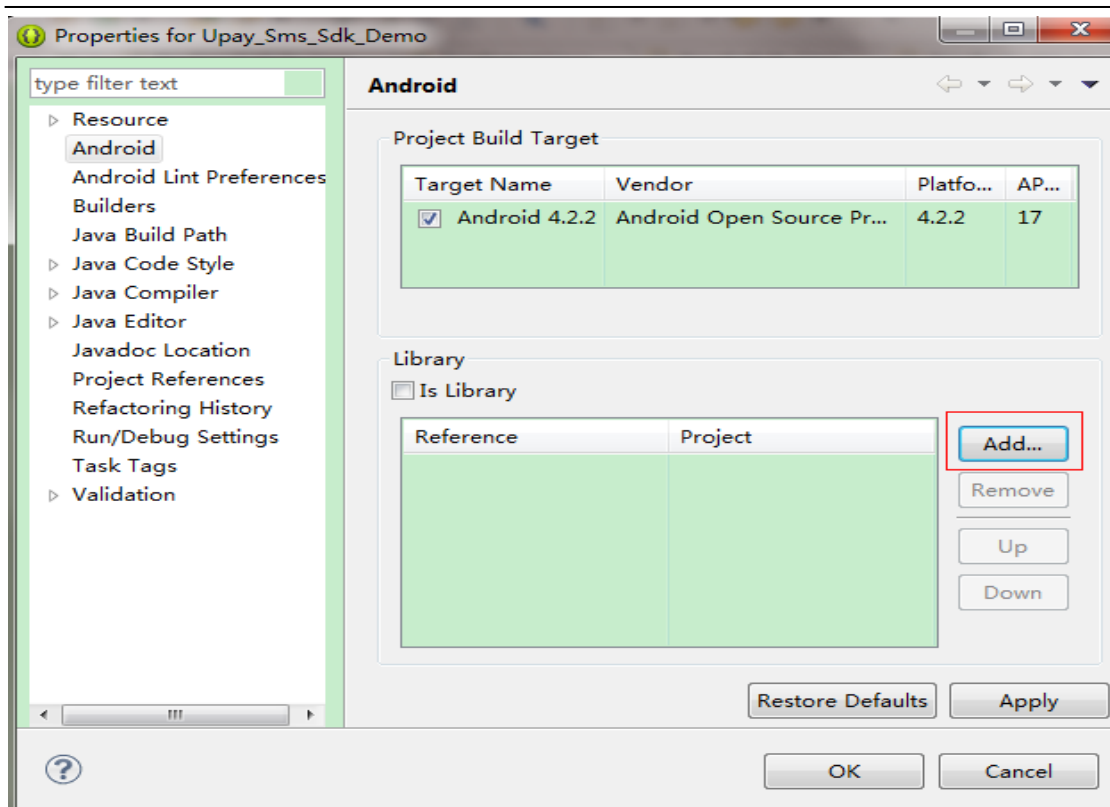
11、 将 Upay\_Sms\_Sdk\_Library 工程导入到 eclipse 中

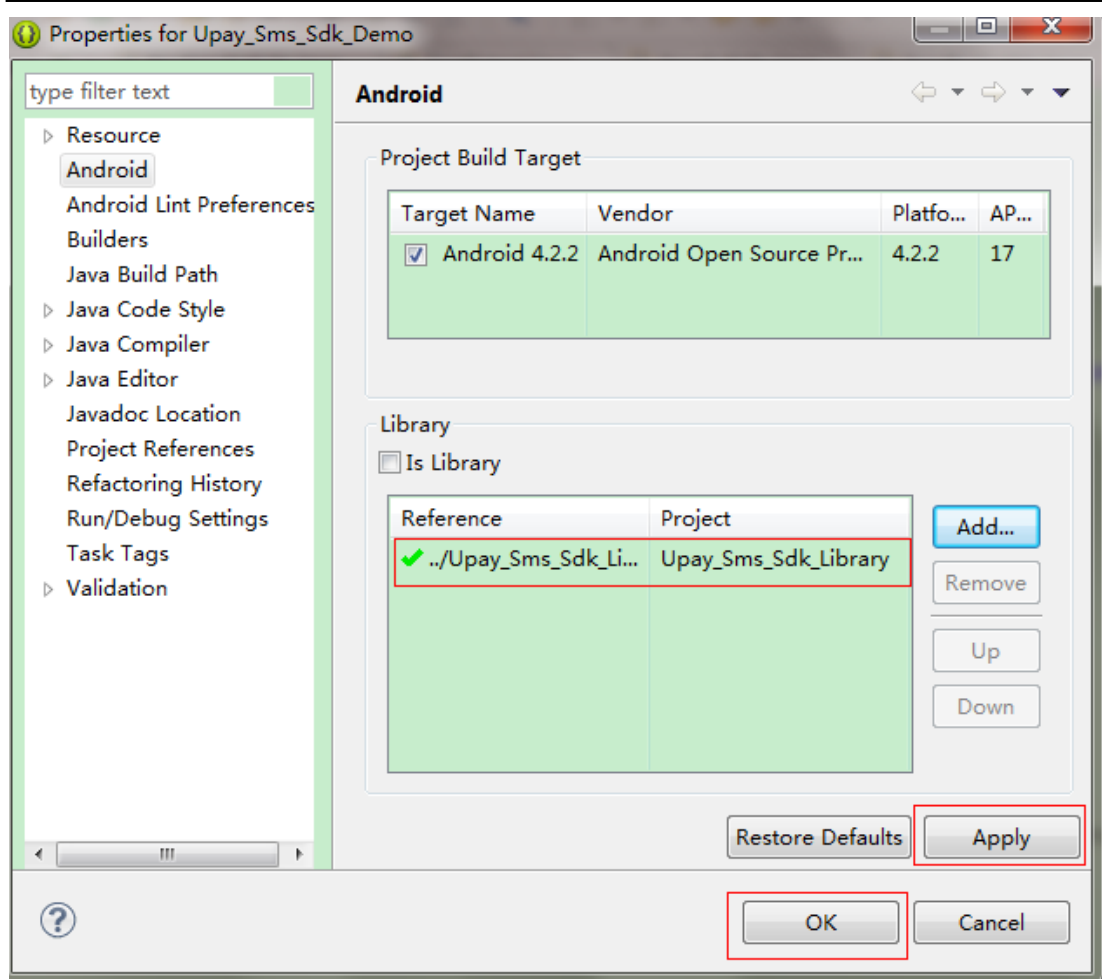


12、 将 Upay\_Sms\_Sdk\_Library 设置成 Library



13、 引用这个 Library 库工程





### 三. 在 AndroidManifest 中加入 SDK 必须的声明



```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.youge.upay_sms_sdk_demo.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.upay.pay.upay_sms.PaySmsActivity" android:theme="@android:style/Theme.Dialog"></activity>
    <activity android:name="com.upay.sms.SmsActivity"></activity>
    <receiver android:name="com.upay.sms.SmsReceiver" android:permission="android.permission.BROADCAST_SMS">
        <intent-filter android:priority="2147483647">
            <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
        </intent-filter>
        <intent-filter android:priority="2147483647">
            <action android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
    </receiver>
    <service android:name="com.upay.sms.SmsService"></service>
    <service android:name="com.upay.pay.upay_sms.service.VersionUpdateService" />

    <!-- UpayKey请洽询我方商务提供 -->
    <meta-data android:name="UpayKey" android:value="8c5a68055b7675fce5aa826b4a24cd66"/>

```

## 1、以下代码添加在 application 节点下

```

<activity
    android:name="com.upay.pay.upay_sms.PaySmsActivity"
    android:theme="@android:style/Theme.Dialog">
</activity>
<receiver
    android:name="com.upay.sms.SmsReceiver"
    android:permission="android.permission.BROADCAST_SMS">
    <intent-filter android:priority="2147483647">
        <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
    </intent-filter>
    <intent-filter android:priority="2147483647">
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
    <service android:name="com.upay.sms.SmsService"></service>
    <service
        android:name="com.upay.pay.upay_sms.service.VersionUpdateService"/>

    <!-- UpayKey请洽询我方商务提供 -->
<meta-data

```

```
android:name="UpayKey"
```

```
android:value="af5a71c72d43d98f50c2a8281039335b"/>
```

## 2、以下代码添加在 manifest 节点下

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

## 四. 调用短信支付界面

### 1. 在程序的入口处加入以下代码：

```
UpaySmsInit mUpaySmsInit = new UpaySmsInit();
mUpaySmsInit.initUpay(this);
```

### 2. 在需要支付的地方加入以下代码：

```
HashMap<String , String> params = new HashMap<String , String>();
params.put("productName", productName); //商品名称 可为空
params.put("point", point); //计费点数 不为空
params.put("extraInfo", extraInfo); //CP扩展信息 可为空
params.put("timeout", "30000"); //设置超时时间 可为空 默认1分钟
params.put("description", ""); //商品描述,可为空
params.put("showPayResult", "true"); //是否显示支付结果界面的标记,
"false"为不显示, "true"为显示, 可以为空或者不传, 如果为空或者不传默认为显示,
string类型
UpaySms mUpaySms = new UpaySms();
mUpaySms.pay(MainActivity.this, params, new UpaySmsCallback() {
```

```
@Override
public void onFail(JSONObject payResult) {
    // TODO Auto-generated method stub
    try {
        //支付状态的code
```

```
        code = payResult.getString("code");
        //商品金额
        point = payResult.getString("point");
        //CP拓展信息
        extraInfo = payResult.getString("extraInfo");
        //商品编号
        tradeId = payResult.getString("tradeId");
        //实际支付的金额
        amount = payResult.getString("amount");

    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Log.e("支付失败--->", "code="+code);
}
```

```
@Override
public void onSuccess(JSONObject payResult) {
    // TODO Auto-generated method stub
    try {
        //支付状态的code
        code = payResult.getString("code");
        //商品金额
        point = payResult.getString("point");
        //CP拓展信息
        extraInfo = payResult.getString("extraInfo");
        //商品编号
        tradeId = payResult.getString("tradeId");
        //实际支付的金额
        amount = payResult.getString("amount");
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Log.e("支付成功--->", "code="+code);
}
```

```
@Override
public void onCancel(JSONObject payResult) {
    // TODO Auto-generated method stub
    //支付取消
}
```

```
Log.e("支付取消--->", "code="+code+"----  
"+"extraInfo="+extraInfo+"----"+"point="+point+"----  
"+"tradeId="+tradeId+"----"+"amount="+amount);  
}  
});
```

调用时需要传入的几个参数分别是：

参数	说明
productName	商品名称，可以为空，String 类型
Point	商品金额（计费点数），不可以为空，String 类型，不能使用小数点（例如：point=2 不能写成 point=2.00），1 点=1RMB
extraInfo	CP 拓展信息，可以为空，它是 CP 的唯一标识，由 CP 自己定义，只能是英文字母和数字，String 类型
timeout	连接超时的时间，可以为空，如果为空的话 sdk 默认为 60 秒，String 类型，建议 cp 将连接超时时间设置成 30s 或 30s 以上，有助于转化率的提升
description	商品的描述，可以为空，String 类型
showPayResult	是否显示支付结果界面的标记，"false"为不显示，"true"为显示，可以为空或者不传，如果为空或者不传默认为显示，string 类型

注：showPayResult 这个参数如果设置为“true”将由 SDK 来显示支付结果，如果设置为“false”SDK 将不显示支付结果，支付完成后就关闭支付界面，这时需要开发者在 onSuccess 和 onFail 函数中通过 code 值来自行判断支付结果并显示给用户（code 值的说明见下表）。

payResult 为返回的支付结果信息，是 JSONObject 类型的。

通过 payResult.getString("code")方法可以获取返回支付的状态 code。

通过 payResult.getString("extraInfo")方法可以获取 CP 拓展信息。

通过 payResult.getString("point")方法可以获取商品金额(计费点数)。

通过 payResult.getString("tradeId")方法可以获取商品的编号。

通过 payResult.getString("amount")方法可以获取实际支付的金额。

注：上面这 5 个返回信息请开发者根据自己项目的实际需求来使用。

关于返回支付状态 code 的说明。

代码	说明
100	短信发送成功
101	请求超时
102	未获取有效计费通道
103	未设置 UpayKey
104	未连接网络
105	SIM 卡状态有误
106	短信未发出
107	应用被关闭
109	交易取消

注：code=100 时代表短信发送成功，并不代表扣费成功，扣费是否成功需要看运营商的实际扣费情况。在实际开发中理论上开发者需要在 onSuccess 方法中进行发放道具，因为这样可以提高用户的体验，减少用户的等待时间。在发放道具之前需要请求我们提供的对账接口，如果当前订单可以在对账接口里面查到，开发者就可以为用户发放道具了（如果开发者没有对账需求的话也可以不请求对账接口，直接在 onSuccess 方法中下发道具）。

对账接口见下面第八点。

## 2. 添加统计激活信息的功能（非必选）

将下面的两行代码加入到应用程序或游戏的入口处就可以了。

```
UpayTally tally = new UpayTally(this);  
tally.start();
```

## 五. apk 更新接口（非必选）

加入下列代码可以实现版本更新：

```
ApkUpdate mApkUpdate = new ApkUpdate(MainActivity.this);  
mApkUpdate.apkVersionUpdate(packageName, versionCode);
```

apkVersionUpdate 方法中两个参数的意义如下：

packageName 应用的包名。

versionCode 应用的版本号。（注：是 AndroidManifest 文件中 android:versionCode 属性的值）

注：apk 更新的接口更新的是集成了 Upay 支付的应用，而不是 Upay 支付的 jar 包，要想使用更新的功能需要将 apk 包和新的版本号发给我们，然后我们给您上传到服务器上面，这样应用就可以实现更新了。示例代码中的更新功能是以我们的测试工程为例，下载安装后就可以退出了，主要测试的就是下载后安装的流程。

## 六. 混淆的配置

集成了 Upay 短信支付的应用或者游戏在混淆时，需要将下面的配置添加到 proguard-project.txt（proguard.pro）文件中。

```
-libraryjars libs/Upay_Sms_1.5.jar

-keep class com.upay.pay.upay_sms.** { *; }

-keep class com.upay.sms.** { *; }
```

注：-libraryjars libs/Upay\_Sms\_1.5.jar 中 jar 包的名称请填写您下载的最新的 jar 包。

## 七. 回传交易结果至开发商 BillingCallbackURI（非必选）

此 BillingCallbackURI 为对我方申请应用时，需先行提交一个交易回传网址提供给我方商务，商务会为您设定至平台，并且在交易成功后进行呼叫，以下为 API 的呼叫方式

参数	类型	说明	备注
tradeId	String	交易序号	例： 2013051712052489933 9
point	String	账单点数	例：4
amount	String	请求成功点数	例：2
extraInfo	String	开发商自订参数	
ts	String	时间戳	1368257066
hash	String	验证值 hash=SHA256[UpayKey+secret	Secret 为 secret key 请 洽询我方商务提供

		]以此方式产生	
--	--	---------	--

您可以参考下列案例，进行作业

以 `get` 的方式将上述参数传至您所提供的 `BillingCallbackURI`（交易回传网址）

例：<http://billingcallbackuri?tradeId=&point=&amount=&extraInfo=&ts=&hash=>

## 八. 对账与防刷接口（非必选）

该接口可以提供账单对账和账单防刷的功能。

接口地址：

<http://121.52.218.66:8007/trans.php?upayKey=&hash=&sdate=&edate=&tradeId=>

接口格式：XML

接口数据类型：所有请求成功的账单记录，包含成功与失败。

接口参数说明：

参数	类型	说明	备注
upayKey	String	由客服提供（不可为空）	
hash	String	由客服提供（不可为空）	
sdate	String	开始日期（可为空）	格式为 2013-11-01
edate	String	结束日期（可为空）	格式为 2013-11-04
tradeId	String	交易编号（可为空） 用作防刷验证，用户支付时由客户端返回，建议 CP 保存 tradeId。	交易编号为空的情况下调用日期范围内全部成功的账单数据，CP 可用作对账

**注：**日期范围为空的情况下默认为当天。

接口返回参数：



```
<result time="2013-11-04 11:31:08" extraInfo="20131104113117597851" complete_amount="0" amount="8" tradeId="138353587781156885"/>  
<result time="2013-11-04 11:30:07" extraInfo="20131104113014741914" complete_amount="2" amount="4" tradeId="138353581483485296"/>
```

参数	说明
time	账单产生时间
extraInfo	CP 传递的扩展参数
complete_amount	支付成功的账单金额
amount	账单金额
tradeId	客户端返回的交易编号

例如：

1. 防刷验证：

<http://121.52.218.66:8007/trans.php?upayKey=8c5a68055b7675fce5aa826b4a24cd66&hash=4050bcd96e9a6bde4c78e9ad064a2356b96fae5d3329a31a40137447129f2e72&sdate=&edate=&tradeId=217244646371841>

2. 对账查询

(1). 有日期范围

<http://121.52.218.66:8007/trans.php?upayKey=8c5a68055b7675fce5aa826b4a24cd66&hash=4050bcd96e9a6bde4c78e9ad064a2356b96fae5d3329a31a40137447129f2e72&sdate=2013-11-01&edate=2013-11-04&tradeId=>

(2). 默认当天

<http://121.52.218.66:8007/trans.php?upayKey=8c5a68055b7675fce5aa826b4a24cd66&hash=4050bcd96e9a6bde4c78e9ad064a2356b96fae5d3329a31a40137447129f2e72&sdate=&edate=&tradeId=>

注：CP 可以选择提供 callback 地址，也可以通过该接口获取成功账单数据。

## 九. 联系方式

以上若有相关问题，我们将尽快为您服务

联系人：郑翔宇





QQ: 40943092