

# Android UpaySDK 使用文档

## 目录

|                                    |    |
|------------------------------------|----|
| 1、SDK 简介                           | 2  |
| 1.1 前言                             | 2  |
| 1.2 主要功能                           | 2  |
| 1.3 框架设计                           | 2  |
| 1.4 工作模式                           | 2  |
| 2、开发指南                             | 2  |
| 3、集成步骤                             | 2  |
| 3.1、Eclipse 下集成 SDK                | 2  |
| 3.2、Android studio 下集成 SDK         | 4  |
| 4、重要参数说明                           | 4  |
| 4.1、UpayTask 类                     | 4  |
| 4.2、UpayOrder 类说明                  | 5  |
| 4.3、UpayResult 类说明                 | 7  |
| 4.4、UpayResult 中 result_code 状态码说明 | 8  |
| 5、SDK 有界面模式调用示例                    | 8  |
| 5.1、激活                             | 8  |
| 5.2、支付                             | 9  |
| 5.3、退款                             | 9  |
| 5.4、撤单                             | 9  |
| 5.5、预下单                            | 10 |
| 6、SDK 无界面模式调用示例                    | 10 |
| 6.1、激活                             | 10 |
| 6.2、支付                             | 10 |
| 6.3、退款                             | 11 |
| 6.4、查询                             | 11 |
| 6.5、撤单                             | 11 |
| 6.6、预下单                            | 11 |
| 7、联系我们                             | 12 |
| 8、版本记录                             | 12 |

## 1、SDK 简介

### 1.1 前言

收钱吧 Android SDK2.0 集成了各种支付方式，第三方通过调用本 SDK，可以轻松将支付宝，微信，百度钱包，京东钱包支付等多种支付方式集成到自己的应用。

### 1.2 主要功能

- (1)激活：新的终端使用 SDK，需要先激活终端，才可以进行交易
- (2)付款：支持支付宝、微信、百度钱包、京东钱包支付
- (3)退款：对已经支付成功的订单进行退款操作
- (4)查询：对已经支付的订单查询
- (5)撤单：直接取消某一笔交易
- (6)预下单：支持支付宝、微信、百度钱包、京东钱包支付

### 1.3 框架设计

第三方用户通过集成收钱吧 SDK，面对前方商户提供多种可选的支付方式，同时喔嚒可以将交易路由至支付宝，微信等服务器。

### 1.4 工作模式

#### 1.4.1、SDK 标准界面模式

SDK 默认的工作模式，激活、付款、退款、撤单、预下单集成了 UI 界面，不需要特别配置。

#### 1.4.2、SDK 无界面标准模式

SDK 无界面的工作模式，在调用接口时，需要在 UpayOrder 订单参数中指定调用模式为 setPayModel(UpayOrder.PayModel.NO\_UI)。

## 2、开发指南

开发包获取：去官方网站获取最新开发包 <https://shouqianba.com/sdkDocs.php?title=Android>。

## 3、集成步骤

### 3.1、Eclipse 下集成 SDK

#### 3.1.1、解压 SDK，得到以下文件夹：

|         |                |     |
|---------|----------------|-----|
| demo    | 2016/1/6 15:41 | 文件夹 |
| eclipse | 2016/1/6 15:43 | 文件夹 |
| studio  | 2016/1/6 15:41 | 文件夹 |

#### 3.1.2、打开 eclipse 文件夹，将文件夹下面的 libs,与 res 文件夹 copy 到集成的项目中；

|      |                 |     |
|------|-----------------|-----|
| libs | 2016/1/12 11:50 | 文件夹 |
| res  | 2016/1/6 15:55  | 文件夹 |

#### 3.1.3、在 AndroidManifest.xml 中声明以下权限；

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<!-- 允许程序读取或写入系统设置 -->
```

```

<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<!-- GPS 定位权限 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<!-- SD 卡读取权限，用户写入离线定位数据-->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.REORDER_TASKS" />

```

3.1.4、在 AndroidManifest.xml 中声明以下 SDK 用到的 activity、service、receiver;

```

<activity
    android:name="com.wosai.upay.ui.MainActivity"
    android:theme="@style/AppBaseTheme"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:windowSoftInputMode="stateHidden|adjustResize">
</activity>
<activity
    android:name="com.wosai.upay.zbar.activity.CaptureActivity"
    android:theme="@style/AppBaseTheme"
    android:windowSoftInputMode="adjustPan">
</activity>
<receiver
    android:name="com.wosai.upay.common.UpayTask$MyBroadcastReceiver"
    android:exported="false">
    <intent-filter>
        <action android:name="com.wosai.upay" />
    </intent-filter>
</receiver>
<meta-data
    android:name="com.baidu.lbsapi.API_KEY"
    android:value="jb5QfTeGACZiPIUBx9ZwIC82" />
<service android:name="com.baidu.location.f" android:enabled="true"
    android:process=":remote">
</service>

```

3.1.5、初始化 SDK，在程序入口处初始化 SDK，如果初始化提示 AndroidManifest.xml 文件配置有误，请检查配置文件;

```

/**
 * initUpay 初始化 UpayTask
 * @param context 上下文对象
 * @param isPlaySound 是否打开交易完成时，成功和失败的提示声音
 * @param urlType 服务器地址类型：DEV：开发环境，PRO：生产环境
 */
DebugConfig.setDebug(true); //默认为非调试模式，如果需要调试，请设置为 true，打印和保存相关日志
UpayTask.getInstance().initUpay(context, isPlaySound, Env.UrlType.DEV);

```

3.1.6、成功集成 SDK 到项目中.

## 3.2、Android studio 下集成 SDK

3.2.1、解压 SDK , 得到以下文件夹;

|         |                |     |
|---------|----------------|-----|
| demo    | 2016/1/6 15:41 | 文件夹 |
| eclipse | 2016/1/6 15:43 | 文件夹 |
| studio  | 2016/1/6 15:41 | 文件夹 |

3.2.2、打开 studio 文件夹 , 拷贝 SQB-Android-SDK-x.x.x.aar 结尾的压缩包到项目的 libs 下;



3.2.3、在 build.gradle 文件中添加以下代码;

```
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    compile(name: 'SQB-Android-SDK-2.1.0', ext: 'aar')//name 后面紧跟的为引入的 aar 包名
}
```

3.2.4、初始化 SDK , 在程序入口处初始化 SDK , 如果初始化提示 AndroidManifest.xml 文件配置有误, 请检查配置文件;

```
/**
 * initUpay 初始化 UpayTask
 * @param context 上下文对象
 * @param isPlaySound 是否打开交易完成时, 成功和失败的提示声音
 * @param urlType 服务器地址类型: DEV: 开发环境, PRO: 生产环境
 */
DebugConfig.setDebug(true); //默认为非调试模式, 如果需要调试, 请设置为 true, 打印和保存相关日志
UpayTask.getInstance().initUpay(context, isPlaySound, Env.UrlType.DEV);
```

3.2.5、SDK 成功集成到项目.

## 4、重要参数说明

### 4.1、UpayTask 类

4.1.1、UpayTask 对象获取

UpayTask.getInstance()

4.1.2、UpayTask 方法说明

```
/**
 * activate 激活
 * @param code 激活码内容
```

```

    * @param vendor_id 服务商ID
    * @param vendor_key 服务商KEY
    * @param payModel 支付模式
    * @param callBack 请求回调
    */
    public void activate(String code, String vendor_id, String vendor_key, UpayOrder.PayModel
payModel, UpayCallBack callBack)

/**
    * pay 付款
    * @param order 业务订单类
    * @param callBack 请求回调
    */
    public void pay(UpayOrder order, UpayCallBack callBack)

/**
    * refund 退款
    * @param order 业务订单类
    * @param callBack 请求回调
    */
    public void refund(UpayOrder order, UpayCallBack callBack)

/**
    * query 查询
    * @param order 业务订单类
    * @param callBack 请求回调
    */
    public void query(UpayOrder order, UpayCallBack callBack)

/**
    * revoke 撤单（退货）
    * @param order 业务订单类
    * @param callBack 请求回调
    */
    public void revoke(UpayOrder order, UpayCallBack callBack)

/**
    * preCreate 预下单
    * @param order 业务订单类
    * @param callBack 请求回调
    */
    public void preCreate(UpayOrder order, UpayCallBack callBack)

```

4.2、UpayOrder 类说明

| 属性 | 属性名称 | 类型     | 描述     | 付款 | 退款        | 预下单 | 查询   | 手动撤单      |
|----|------|--------|--------|----|-----------|-----|------|-----------|
| sn | 收钱吧订 | String | 收钱吧系统内 | —— | 无 UI ( sn | ——  | sn 和 | 无 UI ( sn |

|                       |             |                    |  |             |   |     |            |   |
|-----------------------|-------------|--------------------|--|-------------|---|-----|------------|---|
|                       | 单号          |                    | 部唯一订单号   |             | 和   |     | client_sn  | 和   |
| client_sn             | 商户系统<br>订单号 | String             | 必须在商户系<br>统内唯一 ;且长<br>度不超过 32 字<br>节   | 必传          | client_sn<br>不能同时<br>不传 ), 有<br>UI ( sn 和<br>client_sn<br>非必传 ) | 必传  | 不能同时<br>不传 | client_sn<br>不能同时<br>不传 ), 有<br>UI ( sn 和<br>client_sn<br>非必传 ) |
| total_amount          | 交易总金<br>额   | String             | 以分为单位,不<br>超过 10 位纯数<br>字字符串,超过<br>1 亿元的收款请<br>使用银行转账  | 必传          | ——  | 必传  | ——         | ——  |
| payway                | 支付方式        | String             | 纯数字字符串,<br>一旦设置,则根<br>据支付码判断<br>支付通道的逻<br>辑失效。   | 非必传         | ——  | 必传  | ——         | ——  |
| dynamic_id            | 付款码内<br>容   | String             | 不超过 32 字节  | 无 UI 必<br>传 | ——  | ——  | ——         | ——  |
| subject               | 交易简介        | String             | 本次交易的简<br>要介绍  | 必传          | ——  | 必传  | ——         | ——  |
| operator              | 门店操作<br>员   | String             | 发起本次交易<br>的操作员   | 必传          | 必传  | 必传  | ——         | ——  |
| description           | 商品详情        | String             | 对商品或本次<br>交易的描述  | 非必传         | ——  | 非必传 | ——         | ——  |
| extended              | 扩展参数<br>集合  | String             | 收钱吧与特定<br>第三方单独约<br>定的参数集<br>合,json 格式,<br>最多支持 24 个<br>字段,每个字段<br>key 长度不超过<br>64 字节,value<br>长度不超过<br>256 字节 | 非必传         | ——  | 非必传 | ——         | ——  |
| reflect               | 反射参数        | String             | 任何调用者希<br>望原样返回的<br>信息   | 非必传         | 非必传   | 非必传 | ——         | 非必传   |
| refund_request<br>_no | 退款序列<br>号   | String             | 商户退款所需<br>序列号,防止重<br>复退款   | ——          | 必传  | ——  | ——         | ——  |
| refund_amount         | 退款金额        | String             | 退款金额   | ——          | 无 UI 必传   | ——  | ——         | ——  |
| payModel              | 支付模式        | PayModel<br>(enum) | 默认有界面模<br>式  | 非必传         | ——  | 非必传 | ——         | ——  |

|             |      |                   |                         |    |         |    |    |         |
|-------------|------|-------------------|-------------------------|----|---------|----|----|---------|
| refundModel | 退款模式 | RefundModel(enum) | 有界面退款模式,通过商户订单号或者收钱吧订单号 | —— | 有 UI 必传 | —— | —— | ——      |
| revokeModel | 退款模式 | RefundModel(enum) | 有界面退款模式,通过商户订单号或者收钱吧订单号 | —— | ——      | —— | —— | 有 UI 必传 |

4.3、UpayResult 类说明

| 属性                  | 属性名称            | 描述  |
|---------------------|-----------------|---|
| sn                  | 收钱吧订单号          | 收钱吧系统内部唯一订单号  |
| client_sn           | 商户订单号           | 商户系统订单号   |
| trade_no            | 支付服务商订单号        | 支付通道交易凭证号   |
| status              | 流水状态            | 本次操作产生的流水的状态  |
| order_status        | 订单状态            | 当前订单状态  |
| payway              | 支付方式            | 一级支付方式（支付宝、微信、百付宝、京东）                               |
| sub_payway          | 二级支付方式          | 二级支付方式（条码支付、二维码支付）                                  |
| payer_uid           | 付款人 ID          | 支付平台（微信，支付宝）上的付款人 ID                                |
| payer_login         | 付款人账号           | 支付平台上(微信，支付宝)的付款人账号                                 |
| total_amount        | 交易总额            | 本次交易总金额   |
| net_amount          | 实收金额            | 如果没有退款，这个字段等于 total_amount。否则等于 total_amount 减去退款金额 |
| subject             | 交易概述            | 本次交易概述  |
| finish_time         | 付款动作在收钱吧的完成时间   | 时间戳   |
| channel_finish_time | 付款动作在支付服务商的完成时间 | 时间戳   |
| operator            | 操作员             | 门店操作员   |
| description         | 商品详情            | 对商品或本次交易的描述   |
| reflect             | 反射参数            | 透传参数  |
| qr_code             | 二维码内容           | 预下单成功后生成的二维码  |
| result_code         | 业务执行响应码         |   |
| error_code          | 业务执行结果返回码       |   |
| error_message       | 业务执行错误信息        |   |

payway 支付方式列表

| 取值 | 含义  |
|----|-----|
| 1  | 支付宝 |
| 3  | 微信  |
| 4  | 百付宝 |
| 5  | 京东  |

sub\_payway 二级支付方式列表

| 取值 | 含义    |
|----|-------|
| 1  | 条码支付  |
| 2  | 二维码支付 |

4.4、UpayResult 中 result\_code 状态码说明

| 取值                   | 含义                                  | 下一步动作             |
|----------------------|-------------------------------------|-------------------|
| 200                  | 通讯成功                                |                   |
| 400                  | 客户端错误                               |                   |
| 500                  | 服务端错误                               |                   |
| ACTIVATE_SUCCESS     | 激活操作成功                              |                   |
| PAY_SUCCESS          | 支付操作成功                              |                   |
| PAY_FAIL             | 支付操作失败并且已冲正                         | 重新进行一笔交易          |
| PAY_FAIL_ERROR       | 支付操作失败并且不确定第三方支付通道状态                | 联系客服 400-668-5353 |
| CANCEL_SUCCESS       | 撤单操作成功                              |                   |
| CANCEL_ERROR         | 撤单操作失败并且不确定第三方支付通道状态                | 联系客服 400-668-5353 |
| CANCEL_ABORT_ERROR   | 撤单操作试图终止进行中的支付流程，但是失败，不确定第三方支付通道的状态 | 联系客服 400-668-5353 |
| CANCEL_ABORT_SUCCESS | 撤单操作试图终止进行中的支付流程并且成功                |                   |
| REFUND_SUCCESS       | 退款操作成功                              |                   |
| REFUND_ERROR         | 退款操作失败并且不确定第三方支付通状态                 | 联系客服 400-668-5353 |
| PRECREATE_SUCCESS    | 预下单操作成功                             |                   |
| PRECREATE_FAIL       | 预下单操作失败                             |                   |
| SUCCESS              | 操作成功                                |                   |
| FAIL                 | 操作失败（不会触发流程）                        |                   |

如果激活接口 result\_code 返回的不是 ACTIVATE\_SUCCESS，认为激活失败。

如果付款接口 result\_code 返回的不是 PAY\_SUCCESS，认为付款失败。

如果退款接口 result\_code 返回的不是 REFUND\_SUCCESS，认为退款失败。

如果撤单接口 result\_code 返回的不是 CANCEL\_SUCCESS 或 CANCEL\_ABORT\_SUCCESS，认为撤单失败。

如果预下单接口 result\_code 返回的不是 PRECREATE\_SUCCESS，认为预下单失败。

如果查询接口 result\_code 返回的不是 SUCCESS，认为查询失败。

如果 result\_code 返回为 null 或 0，通过 error\_code 和 error\_message 信息处理。

如果 result\_code 返回的是失败响应码，可先通过 error\_code 和 error\_message 信息处理，如果不存在 error\_code 和 error\_message，再根据 result\_code 的含义及下一步处理。

5、SDK 有界面模式调用示例

5.1、激活

```
String vendorId = "10298371039";// 服务商 ID
```



```
String vendorKey = "68d499beda5f72116592f5c527465656";//服务商KEY
UpayTask.getInstance().activate("", vendorId, vendorKey,
UpayOrder.PayModel.UI, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});
```

## 5.2、支付

```
UpayOrder order = new UpayOrder();
order.setClient_sn("client1241241");//商户订单号
order.setTotal_amount("100");//交易总金额
order.setPayway("1");//支付方式
order.setSubject("Android SDK");//交易简介
order.setOperator("操作员 007");//门店操作员
order.setDescription("餐饮");//商品详情
order.setReflect("测试反射参数");//反射参数
UpayTask.getInstance().pay(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});
```

## 5.3、退款

```
UpayOrder order = new UpayOrder();
order.setSn("7894259244086017");//收钱吧订单号
order.setClient_sn("client1241241");//商户订单号
order.setRefund_request_no("23524241");//退款序列号
order.setOperator("操作员 007");//操作员
order.setRefund_amount("100");//退款金额
order.setReflect("测试反射参数");//反射参数
order.setRefundModel(UpayOrder.RefundModel.CLIENT_SN);//指定退款模式为商户订单号退款
UpayTask.getInstance().refund(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});
```

## 5.4、撤单

```
UpayOrder order = new UpayOrder();
order.setSn("7894259244086017");//收钱吧订单号
order.setClient_sn("client1241241");//商户订单号
order.setReflect("测试反射参数");//反射参数
order.setRevokeModel(UpayOrder.RevokeModel.CLIENT_SN);//指定退款模式为商户订单号退款
```

```

UpayTask.getInstance().revoke(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

## 5.5、预下单

```

UpayOrder order = new UpayOrder();
order.setClient_sn("client1241241");//商户订单号
order.setTotal_amount("100");//交易总金额
order.setPayway("1");//支付方式
order.setSubject("Android SDK");//交易简介
order.setOperator("操作员 007");//门店操作员
order.setDescription("餐饮");//商品详情
order.setReflect("测试反射参数");//反射参数
UpayTask.getInstance().preCreate(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

## 6、SDK 无界面模式调用示例

### 6.1、激活

```

String code = "131451574136";//激活码
String vendorId = "10298371039";//服务商 ID
String vendorKey = "68d499beda5f72116592f5c527465656";//服务商 KEY
UpayTask.getInstance().activate(code, vendorId, vendorKey,
UpayOrder.PayModel.NO_UI, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

### 6.2、支付

```

UpayOrder order = new UpayOrder();
order.setClient_sn("client1241241");//商户订单号
order.setTotal_amount("100");//交易总金额
order.setPayway("1");//支付方式
order.setDynamic_id("285362899388441557");//付款码内容
order.setSubject("Android SDK");//交易简介
order.setOperator("操作员 007");//门店操作员
order.setDescription("餐饮");//商品详情
order.setReflect("测试反射参数");//反射参数
order.setPayModel(UpayOrder.PayModel.NO_UI);//指定 SDK 启动模式为无界面模式

```

```

UpayTask.getInstance().pay(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

### 6.3、退款

```

UpayOrder order = new UpayOrder();
order.setSn("7894259244086017");//收钱吧订单号
order.setClient_sn("client1241241");//商户订单号
order.setRefund_request_no("23524241");//退款序列号
order.setOperator("操作员 007");//操作员
order.setRefund_amount("100");//退款金额
order.setReflect("测试反射参数");//反射参数
order.setPayModel(UpayOrder.PayModel.NO_UI);//指定 SDK 启动模式为无界面模式
UpayTask.getInstance().refund(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

### 6.4、查询

```

UpayOrder order = new UpayOrder();
order.setSn("7894259244086017");//收钱吧订单号
order.setClient_sn("client1241241");//商户订单号
UpayTask.getInstance().query(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

### 6.5、撤单

```

UpayOrder order = new UpayOrder();
order.setSn("7894259244086017");//收钱吧订单号
order.setClient_sn("client1241241");//商户订单号
order.setReflect("测试反射参数");//反射参数
order.setPayModel(UpayOrder.PayModel.NO_UI);//指定 SDK 启动模式为无界面模式
UpayTask.getInstance().revoke(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});

```

### 6.6、预下单

```
UpayOrder order = new UpayOrder();
order.setClient_sn("client1241241");//商户订单号
order.setTotal_amount("100");//交易总金额
order.setPayway("1");//支付方式
order.setSubject("Android SDK");//交易简介
order.setOperator("操作员 007");//门店操作员
order.setDescription("餐饮");//商品详情
order.setReflect("测试反射参数");//反射参数
order.setPayModel(UpayOrder.PayModel.NO_UI);//指定 SDK 启动模式为无界面模式
UpayTask.getInstance().preCreate(order, new UpayCallBack() {
    @Override
    public void onExecuteResult(UpayResult result) {
        //回调结果，请参考 UpayResult 类说明，根据回调结果添加自己的业务逻辑的处理
    }
});
```

7、联系我们

如以上信息无法帮助你解决在开发中遇到的问题，请通过以下方式联系我们  
邮箱：sdk@wosai-inc.com  
我们的工程师会在第一时间回复您。

8、版本记录

| 版本     | 日期        | 作者  | 说明                            |
|--------|-----------|-----|-------------------------------|
| V2.1.0 | 2016/1/25 | 叶金新 | 初稿                            |
| V2.1.0 | 2016/1/28 | 叶金新 | 更新激活接口，增加 revoke 接口说明         |
| V2.1.1 | 2016/5/4  | 叶金新 | AndroidManifest.xml 删除重复的权限声明 |