

## 现在支付 跨境支付宝安卓插件 接入指南 V 1.0.0

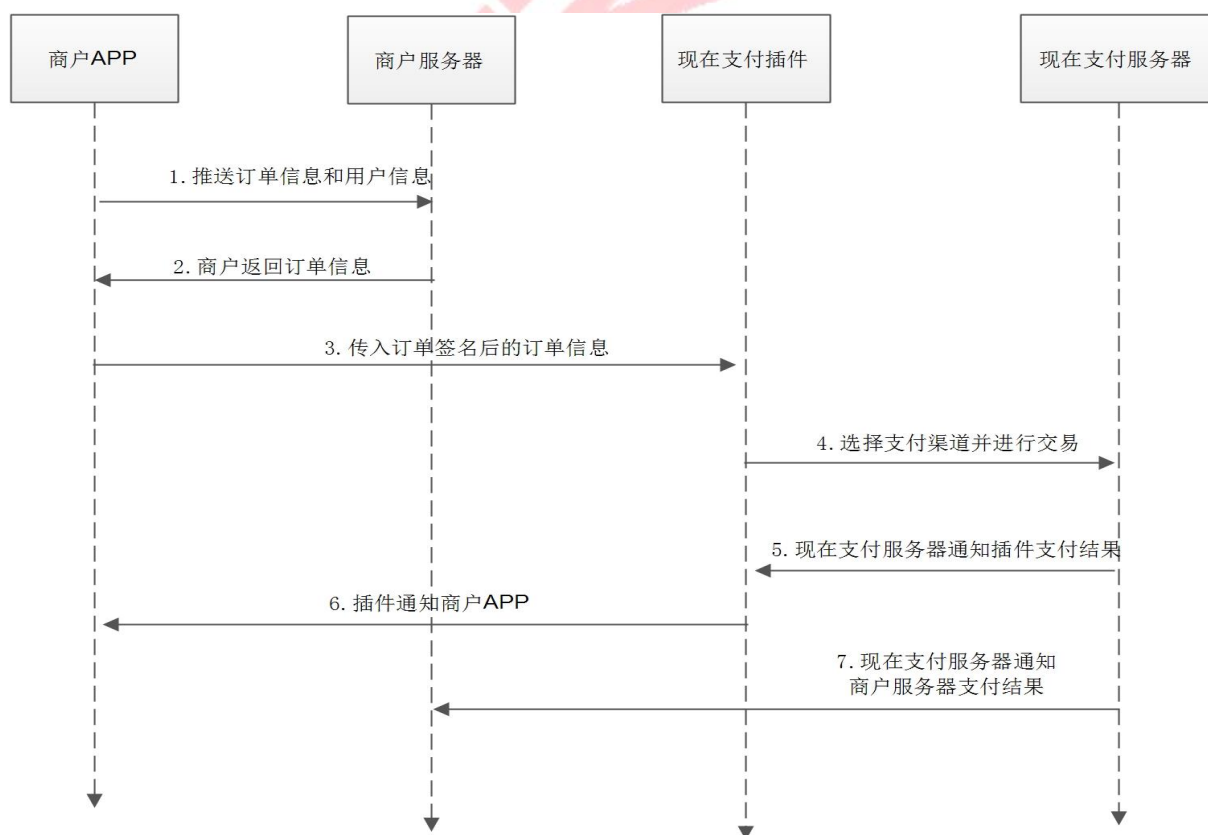
## 目录

一、概述.....	3
1.1 支付流程介绍:.....	3
二、Android 客户端接入.....	4
3.1 接入所需包介绍:(支持单独调起插件).....	5
3.2 插件接口介绍:.....	6
附录 A.....	9
附录 B.....	10
附录 C.....	11
附录 D.....	12

## 一、概述

现在支付跨境控件主要为开发者的手机客户端提供安全、便捷的支付服务，目前支付控件支持 Android 和 iOS 两个平台，用户通过输入支付宝账号等有效信息完成支付。

### 1.1 支付流程介绍:



步骤说明:

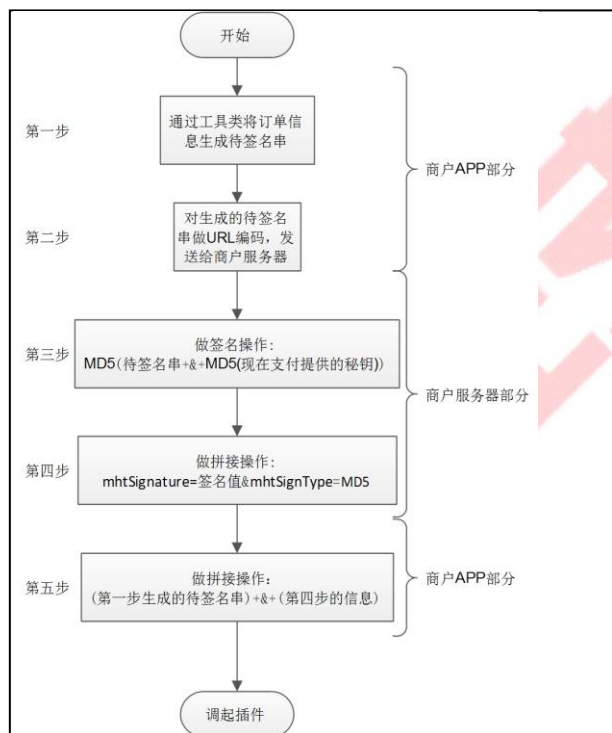
1. 商户 APP 向商户服务器发送订单信息以及账户信息。
2. 商户 APP 将符合插件调起接口规范的信息传入调起方法，并调起现在支付支付插件。
3. 用户在插件中支付渠道的选择并完成支付操作。
4. 支付完成后，现在支付支付插件接收服务器发送的支付结果通知。
5. 现在支付支付插件通知商户 APP 支付情况。(交易状态以商户后台收到的支付结果通知为准)
6. 支付成功后，现在支付服务器通知商户服务器交易信息。

注意:

为了安全考虑, 推荐商户服务器收到信息后, 根据插件调起接口规范说明 (见附录 B) 组合信息, 并对指定字段进行 MD5 签名。

## 二、Android 客户端接入

android 版简便接入流程介绍:



详细介绍:

第一步: 使用插件中的 `PreSignMessageUtil` 工具类生成待签名方法:

对类中的字段进行复制, 调用 `generatePreSignMessage()` 方法进行待签名串的生成。

(若生成结果为 `null`, 则说明有必传参数没有赋值)

```
preSign.appId = mAppIdET.getText().toString();
preSign.mhtOrderNo = mhtOrderNo;
preSign.mhtOrderName = mOrderNameET.getText().toString();
preSign.mhtOrderAmt = mAmtET.getText().toString();
preSign.mhtOrderDetail = mOrderDatailET.getText().toString();
preSign.mhtOrderStartTime = mhtOrderStartTime;
preSign.mhtReserved = mReservedET.getText().toString();
preSign.notifyUrl = mUrlET.getText().toString();
preSign.mhtOrderType = "01";
preSign.mhtOrderTimeOut = "3600";
preSign.mhtCharset = "UTF-8";
preSign.payChannelType = "90";
preSign.mhtCurrencyType = "USD";
preSign.mhtAmtCurrFlag = "1";
```

```
// 生成待签名串
preSignStr = preSign.generatePreSignMessage();
```

第二步:使用插件中的 PluginTools 类的 urlEncode()方法对待签名串进行 urlEncode 编码。

```
GetMessage gM = new GetMessage();  
gM.execute("paydata="+MerchantTools.urlEncode(preSignStr));
```

第三步与第四步见后台样例代码（后台服务器接收到数据需要先做 UTF-8 的 url 解码再进行签名）

第五步:第一步生成的待签名串与第四步服务器生成的签名串拼接起来，传入插件调起方法中。

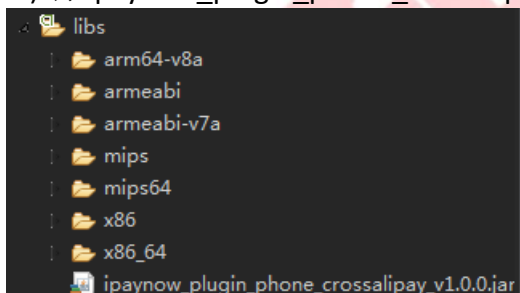
```
String needcheckmsg = HttpUtil.post(GETORDERMESSAGE_URL, msg);  
needcheckmsg=MainActivity.preSignStr+"&"+needcheckmsg;
```

## 3.1 接入所需包介绍:(支持单独调起插件)

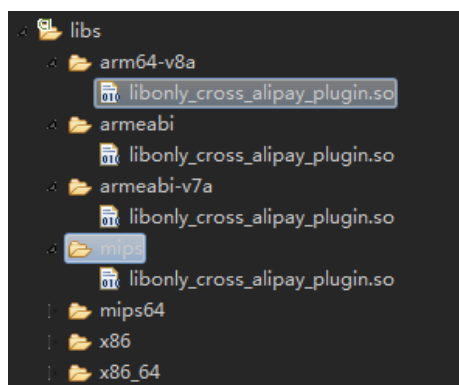
1. ipaynow\_plugin\_phone\_crossalipay\_v1.0.0.jar（现在支付 jar 包）
2. libonly\_cross\_alipay\_plugin.so (现在支付 so 文件)

接入步骤介绍:

a).将 ipaynow\_plugin\_phone\_crossalipay\_v1.0.0.jar 文件拖进项目的 libs 文件夹中



c).将 libonly\_cross\_alipay\_plugin.so 文件夹拖进 libs 文件夹下的 xxx 目录下,其中 xxx 为 armeabi,armeabi-v7a,mips,x86 之一。



## 3.2 插件接口介绍:

a).调起插件支付接口:(**必须在主线程调用，子线程调用的话没动静哟**)

CrossAliPayPlugin.getInstance()

.pay(String req)

该接口需要传入参数为:

req:经过商户服务器签名后的符合现在支付接口规范的请求信息。

```
CrossAliPayPlugin.getInstance()  
    .setCallResultReceiver(MainActivity.this)// 传入继承了通知接口的类  
    .pay(result, MainActivity.this);// 传入请求数据
```

具体 requestMessage 传入信息见附录 A

b)接收通知:

第一步: 调用.setCallResultReceiver(MainActivity.this)//传入继承了通知接口的类

```
public class MainActivity extends Activity implements ReceivePayResult
```

第二步: 在接收通知的类中实现 ReceivePayResult 接口中的 onIpaynowTransResult 方法

```
@Override  
public void onIpaynowTransResult(ResponseParams resp) {  
    String respCode = resp.respCode;  
    String errorCode=resp.errorCode;  
    String respMsg =resp.respMsg;  
    StringBuilder temp=new StringBuilder();  
    if (respCode.equals("00")) {  
        temp.append("交易状态:成功");  
    }else if(respCode.equals("02")) {  
        temp.append("交易状态:取消");  
    }else if(respCode.equals("01")) {  
        temp.append("交易状态:失败").append("\n").  
            append("错误码:").append(errorCode).  
            append("原因:"+respMsg);  
    }else if(respCode.equals("03")) {  
        temp.append("交易状态:未知").append("\n").  
            append("错误码:").append(errorCode).  
            append("原因:"+respMsg);  
    }  
    Toast.makeText(this, temp.toString(), Toast.LENGTH_LONG).show();  
}
```

插件通知信息详见附录 C 列表

## c) 监听 Activity 销毁:

```
@Override
protected void onDestroy() {
    super.onDestroy();
    WechatPayPlugin.getInstance().onActivityDestroy();
}
```

### 特殊说明:

#### 1. 签名部分:

推荐使用插件中的 **PreSignMessageUtil** 类进行待签名串的生成，只需将该类的属性赋值，并调用 **generatePreSignMessage()** 方法，即可生成符合规范的待签名串。客户端需要将该工具类生成的串发送至商户服务器进行 **MD5** 签名(详见附录)。

#### 2. 单独调起插件部分:

若需要跳过网关，直接调起插件，则需要将 **payChannelType** 赋值为指定渠道的标识即可。



注意:

**Activity 配置:**

```
<activity
    android:name="com.alipay.sdk.app.H5PayActivity"
    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
    android:exported="false"
    android:screenOrientation="behind"
    android:windowSoftInputMode="adjustResize|stateHidden" >
</activity>
```

**权限配置:**

```
<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.READ_PHONE_STATE" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

**代码混淆:**

```
-keep public class com.ipaynow.alipay.plugin.utils.PreSignMessageUtil{
    <fields>;
    <methods>;
}
-keep public class com.ipaynow.alipay.plugin.api.CrossAliPayPlugin{
    <fields>;
    <methods>;
}
-keep public class
com.ipaynow.alipay.plugin.manager.route.dto.RequestParams{
    <fields>;
    <methods>;
}
-keep public class
com.ipaynow.alipay.plugin.manager.route.dto.ResponseParams{
    <fields>;
    <methods>;
}
```



## 附录 A

调起插件接口信息规范:

字段名称	字段 Key	格式	必填	备注
商户应用唯一标识	appld	String(1,40)	Y	
商户订单号	mhtOrderNo	String(1,40)	Y	
商户商品名称	mhtOrderName	String(1,40)	Y	
商户交易类型	mhtOrderType	String(1,40)	Y	01 普通消费
商户申请结算币种类型	mhtCurrencyType	String (3)	Y	参见附录 D 支持币种列表
商户订单交易金额	mhtOrderAmt	Number(22)	Y	单位: 分 整数, 无小数点
金额币种单位标记	mhtAmtCurrFlag	String(2)	Y	0: 订单金额单位为人民币 1: 商户申请的结算币种类型
商户订单详情	mhtOrderDetail	String(0,200)	Y	
商户订单超时时间	mhtOrderTimeOut	Number(4)	N	60~3600 秒, 默认 3600
商户订单开始时间	mhtOrderStartTime	String (14)	Y	yyyyMMddHHmmss
商户后台通知 URL	notifyUrl	String(1,200)	Y	HTTP 协议或者 HTTPS 协议, POST 方式提交报文
商户字符编码	mhtCharset	String(1,16)	Y	UTF-8
渠道类型	payChannelType	String(2)	Y	90:跨境支付宝
商户保留域	mhtReserved	String(100)	N	给商户使用的字段, 商户可以对交易进行标记, 现在支付将原样返回
商户数据签名	mhtSignature	String(1,64)	Y	签名逻辑见接口附录说明 5.1 BXXX 交易的 MD5 签名逻辑说明。除如下字段外, 其它字段都参与 MD5 签名。排除的有: mhtSignType, mhtSignature

## 附录 B

第一步：对参与 MD5 签名的字段按字典升序排序后，分别取值后并排除值为空的字段键值对，最后组成 key1=value1&key2=value2....keyn=valuen "表单字符串"。

第二步：对 MD5 密钥进行加密得到"密钥 MD5 值"。

第三步：最后对 第一步中得到的表单字符串&第二步得到的密钥 MD5 值 做 MD5 签名

PS：MD5 密钥是用户在注册应用的时候生成的，每个应用一个 MD5 密钥。

样例：

```
appId=8888888888888888&mhtCharset=UTF-8&mhtOrderNo=20140821161747&mhtOrderName=%E9%99%B6%E6%A0%91%E5%BC%BA&mhtOrderType=01&mhtCurrencyType=156&mhtOrderAmt=1&mhtOrderDetail=%E5%85%B3%E4%BA%8E%E8%AE%A2%E5%8D%95%E9%AA%8C%E8%AF%81%E6%8E%A5%E5%8F%A3%E7%9A%84%E6%B5%8B%E8%AF%95&mhtOrderStartTime=20140821161747&notifyUrl=http%3A%2F%2Flocalhost%3A10802%2F&mhtSignature=72e3b9fea03b81b88224fe0eab1459d9&mhtSignType=MD5
```

### 使用简便流程注意：

如用安卓文档中推荐的简便接入流程的话，服务器只要根据商户 APP 上送的待签名字符串根据以下公式生成签名值即可：

签名公式：**MD5**（待签名串+"&"**MD5**（现在支付提供的密钥））；

生成签名值后，根据以下公式拼接好发送给商户 APP：

拼接公式：**mhtSignature**=（签名公式得到的值）+"&" **mhtSignType**=MD5；

样例：**mhtSignature**=1519adb35c04e0b962d8ca68476d9d56&**mhtSignType**=MD5

商户使用简便流程时，商户后台需要先做 **UTF-8** 的 **url** 解码后再做签名处理

## 附录 C

插件通知接口信息说明:

字段名称	字段 Key	必填	备注
响应码	respCode	Y	交易状态成功:00 交易状态失败:01 交易状态取消:02 交易状态未知:03
错误码	errorCode	N	未知和失败时会返回,详情见错误码表
响应信息	respMsg	N	未知和失败时会返回,详情见错误码表

errorCode 与 respMsg 说明表:

ErrorCode 标识	对应的 RespMsg 信息
PE010	渠道方其他未知失败
PE002	网络连接超时
PE005	支付插件其他未知错误

## 附录 D

货币符号	货币名称
GBP	英镑
HKD	港币
USD	美元
CHF	瑞士法郎
SGD	新加坡元
SEK	瑞典克朗
DKK	丹麦克朗
NOK	挪威克朗
JPY	日元
CAD	加拿大元
AUD	澳大利亚元
EUR	欧元
NZD	新西兰元
RUB	俄罗斯卢布
MOP	澳门元