

**现在支付**  
**中小开发者商户客户端**  
**接入指南**  
**V 1.0.1**

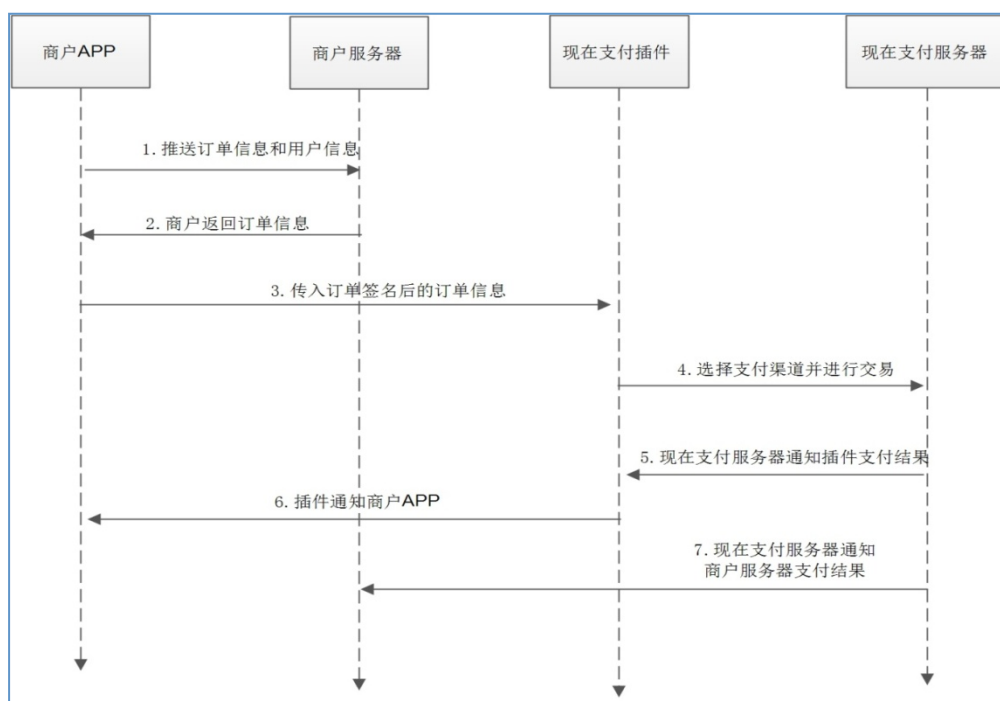
## 目录

一、概述.....	3
1.1 支付流程介绍:.....	3
二、Android 客户端接入（支持横竖屏） .....	4
2.1 接入所需包介绍:.....	5
2.2 插件接口介绍:.....	6
2.3 权限与 Activity 配置 .....	7
三、iOS 客户端接入（支持横竖屏） .....	8
3.1 文件引用 .....	8
3.2 工程设置 .....	8
3.3 调用支付接口 .....	9
附录 A .....	11
附录 B .....	12
附录 C .....	13

## 一、概述

现在支付控件包括银联支付和支付宝支付，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，用户通过输入银行卡号或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:

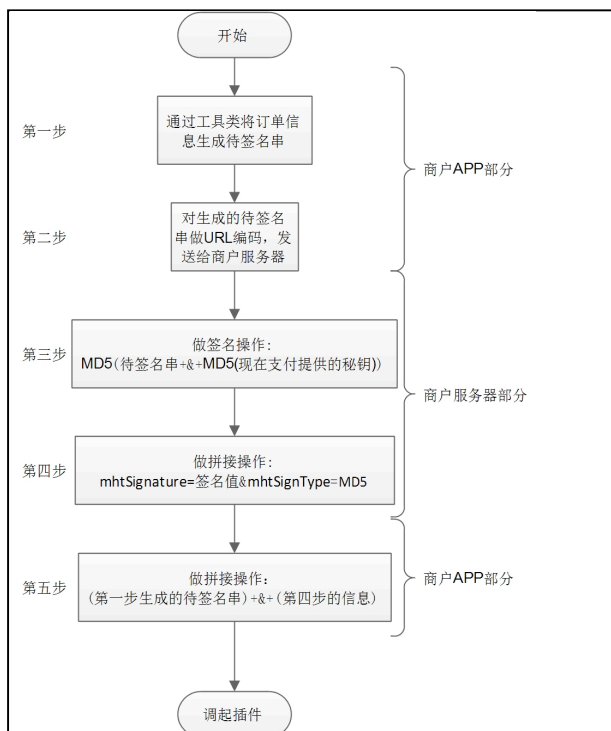
1. 商户 APP 向商户服务器发送订单信息以及账户信息。
2. 商户 APP 将符合插件调起接口规范的信息传入调起方法，并调起现在支付支付插件。
3. 用户在插件中支付渠道的选择并完成支付操作。
4. 支付完成后，现在支付支付插件接收服务器发送的支付结果通知。
5. 现在支付支付插件通知商户 APP 支付情况。**(交易状态以商户后台收到的支付结果通知为准)**
6. 支付成功后，现在支付服务器通知商户服务器交易信息。

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 A）组合信息，并对指定字段进行 MD5 签名。

## 二、Android 客户端接入（支持横竖屏）

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)

```

PreSignMessageUtil preSign=new PreSignMessageUtil();
preSign.appId="1410868994004446";
preSign.mhtOrderNo=new SimpleDateFormat("yyyyMMddHHmmss",Locale.CHINA).format(new Date());
preSign.mhtOrderName="支付样例-手机版";
preSign.mhtOrderType="01";
preSign.mhtCurrencyType="156";
preSign.mhtOrderAmt="100";
preSign.mhtOrderDetail="关于支付的演示";
preSign.mhtOrderTimeOut="3600";
preSign.mhtOrderStartTime=new SimpleDateFormat("yyyyMMddHHmmss",Locale.CHINA).format(new Date());
preSign.notifyUrl="http://localhost:10802/";
preSign.mhtCharset="UTF-8";
preSign.mhtReserved="test";

preSignStr=preSign.generatePreSignMessage();
    
```

(白色字段为选发字段, 若有必选字段没有被设置则返回 `null`)

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

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。

```

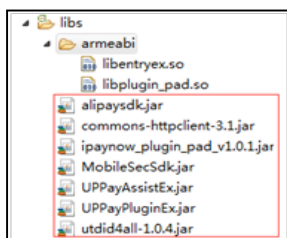
String needcheckmsg = HttpUtil.post(GETORDERMESSAGE_URL, msg);
needcheckmsg=MainActivity.preSignStr+"&"+needcheckmsg;
    
```

## 2.1 接入所需包介绍:

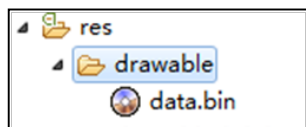
- 1.ipaynow\_plugin.jar (现在支付 jar 包)
- 2.alipaysdk.jar(支付宝 jar 包)
- 3.MobileSecSdk.jar(支付宝 jar 包)
- 4.utdid4all-1.0.4.jar(支付宝 jar 包)
- 5.UPPayPluginEx.jar(银联 jar 包)
- 6.data.bin(银联资源包)
- 7.libentryex.so(银联 so 文件)
- 8.libplugin\_pad.so(现在支付 so 文件)
9. commons-httpclient-3.1.jar

接入步骤介绍:

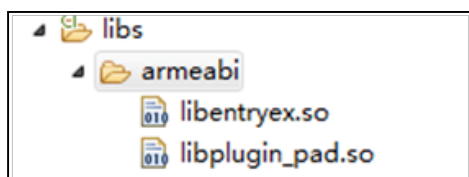
a). 将 ipaynow\_plugin.jar 、 alipaysdk.jar 、 MobileSecSdk.jar 、 utdid4all-1.0.4.jar 、 commons-httpclient-3.1.jar 文件拖进项目的 libs 文件夹中



b).将 data.bin 文件拖进项目 drawable 文件夹中(没有的话就新建一个 drawable 文件夹)。



c).将 libentryex.so、libplugin\_pad.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。



## 2.2 插件接口介绍:

a).调起插件支付接口:

`IpaynowPlugin.pay(Activity,requestMessage);`

该接口需要传入参数为:

`Activity`:商户当前界面的 `Activity` 对象引用。

`requestMessage`: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。

```
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    progressDialog.dismiss();
    IpaynowPlugin.pay(act, result);
}
```

具体 `requestMessage` 传入信息见附录 A

b).插件通知接口:

实现 `Activity` 中的 `onActivityResult(int requestCode, int resultCode, Intent data)`方法。

当用户支付完毕、支付失败、中途取消支付均会通过该方法通知商户 APP。通知时 `requestCode=0`, `resultCode=1`

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (data == null) {
        return;
    }
    String msg = data.getExtras().getString("pay_result");
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("支付结果");
    dialog.setMessage(msg);
    dialog.setInverseBackgroundForced(true);
    dialog.setNegativeButton("确定", new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    dialog.create().show();
}
```

插件通知信息详见附录 C 列表

## 2.3 权限与 Activity 配置

在 sdk 包中找到 AndroidManifest.xml 样例文件。文件有两个，一个是横屏的，另一个是竖屏的，商户可使用其中一种进行配置。

权限如下：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.webkit.permission.PLUGIN" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Activity 配置如下：（以横屏为例）

```
<activity
    android:name="com.ipaynow.plugin.activity.PayMethodActivity"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="landscape"
>
<!-- android:screenOrientation="portrait" -->
<!-- android:screenOrientation="landscape" -->
</activity>
<activity
    android:name="com.ipaynow.plugin.inner_plugin.prepay_plugin.activity.PrePayActivity"
    android:screenOrientation="landscape"
    android:configChanges="orientation">
</activity>
<activity
    android:name="com.ipaynow.plugin.inner_plugin.prepay_plugin.activity.NotifyActivity"
    android:screenOrientation="landscape"
    android:configChanges="orientation">
</activity>
<activity
    android:name="com.unionpay.uppay.PayActivity"
    android:configChanges="orientation|keyboardHidden"
    android:excludeFromRecents="true"
    android:label="@string/app_name"
    android:screenOrientation="landscape">
</activity>
<activity
    android:name="com.unionpay.uppay.PayActivityEx"
    android:configChanges="orientation|keyboardHidden"
    android:excludeFromRecents="true"
    android:label="@string/app_name"
    android:windowSoftInputMode="adjustResize"
    android:screenOrientation="landscape"
>
</activity>
<activity
    android:name="com.alipay.sdk.app.H5PayActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>
<activity
    android:name="com.alipay.sdk.auth.AuthActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>
```

注意：如果您选择竖屏形式，请将图片中红框中的 portrait 改为 landscape 即可。

## 三、iOS 客户端接入（支持横竖屏）

接入步骤介绍:

### 3.1 文件引用

添加 AliPayPlugin 包:

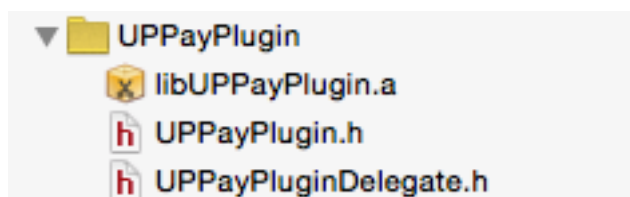
其中包括 AlipaySDK.framework、AlipaySDK.bundle。



拷贝添加 JSON 开源库(SBJSON)源文件(必选)

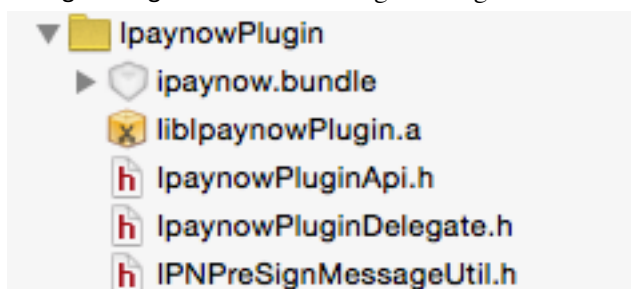
添加 UPPayPlugin 包:

其中包括 libUPPayPlugin.a 、 UPPayPlugin.h 、 UPPayPluginDelegate.h 文件。



添加 IpaynowPlugin 包:

其中包括 ipaynow.bundle 、 libIpaynowPlugin.a 、 ipaynowPluginApi.h 、 ipaynowPluginDelegate.h 及 IPNPreSignMessageUtil.h 文件。



### 3.2 工程设置

在工程的 Build Settings 中找 info,设置 URL Types,添加自定义 URL Scheme





URL Scheme 在回调结果使用,建议起名稍复杂一些,尽量避免同其他程序冲突。

在工程的 Build Settings 中找到 Other Linker Flags 中添加-ObjC 宏。



## 3.3 调用支付接口

**第一步：**使用插件中的 `IPNPreSignMessageUtil` 工具类生成待签名方法: 对类中的字段进行赋值,调用 `generatePreSignMessage()`方法进行待签名串的生成。(若生成结果为 `null`,则说明有必选参数没有赋值)

```
IPNPreSignMessageUtil *preSign=[[IPNPreSignMessageUtil alloc]init];
preSign.appID=@"1410922746865447";
preSign.mhtOrderNo=[dateFormatter stringFromDate:[NSDate date]];
preSign.mhtOrderName=@"手机插件测试用例";
preSign.mhtOrderType=@"01";
preSign.mhtCurrencyType=@"156";
preSign.mhtOrderAmt=@"100";
preSign.mhtOrderDetail=@"关于订单验证接口的测试";
preSign.mhtorderStartTime=[dateFormatter stringFromDate:[NSDate date]];
preSign.notifyUrl=@"http://localhost:10802/";
preSign.mhtCharset=@"UTF-8";
preSign.mhtOrderTimeOut=@"3600";
preSign.mhtReserved=@"test";
preSign.consumerId=@"IPN00001";
preSign.consumerName=@"IpaynowCS";
preSign.payChannelType=@"11";

NSString *originStr=[preSign generatePresignMessage];
```

(`mhtOrderTimeOut`、`mhtReserved`、`consumerId`、`consumerName` 和 `payChannelType` 为选发字段,若有必选字段没有被设置则返回 `nil`)

补充说明: 通过 `payChannelType` 字段可指定跳转到某支付渠道。

第二步：请求后台服务器对待签名串进行签名

```
NSURL* url = [NSURL URLWithString:kSignURL];
NSMutableURLRequest * urlRequest=[NSMutableURLRequest requestWithURL:url];
[urlRequest setHTTPMethod:@"POST"];
urlRequest.HTTPBody=[presignStr dataUsingEncoding:NSUTF8StringEncoding];
NSURLConnection* urlConn = [[NSURLConnection alloc] initWithRequest:urlRequest delegate:self];
[urlConn start];
```

(除 mhtSignature 字段外还需要加入 mhtSignType 字段，示例代码中在后台已处理)

第三步：第一步生成的待签名串与第二步服务器生成的签名串拼接起来,传入插件调起方法中。

```
NSString* data = [[NSMutableString alloc] initWithData:mData encoding:NSUTF8StringEncoding];
NSString* payData=[_presignStr stringByAppendingString:@"&"];
payData=[payData stringByAppendingString:data];
[IpaynowPluginApi pay:payData AndScheme:@"TestPlugin" viewController:self delegate:self];
```

NSString \*payData 主要包含商户的订单信息,key="value"形式,以&连接。

NSString \*scheme 商户程序注册的 URL protocol,供支付完成后回调商户程序使用。

UIViewController\*viewController 商户应用程序调试手机支付的当前 UIViewController。

id<IpaynowPluginDelegate>delegate 实现 IpaynowPluginDelegate 方法的 UIViewController。

第四步：实现插件通知接口

接收通知接口应用内结果通知接口为 IpaynowPluginDelegate，包含如下方法

```
-(void)IpaynowPluginResult:(NSString*)result;
```

控件支付结果将以字符串的形式作为回调函数参数(NSString \*)result 返回。

应用间结果通知接口(用于接收支付宝通知)如下

```
+ (BOOL)handleOpenUrl:(NSURL*)url;
```

在 IpaynowPluginApi.h 中添加 handleOpenUrl:(NSURL\*)url,通过在 AppDelegate.m 中实现来完成独立返回 url 异步通知。

## 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	appId	String(1,40)	Y	现在支付业务提供
商户订单号	mhtOrderNo	String(1,40)	Y	字母、数字
商户商品名称	mhtOrderName	String(1,40)	Y	
商户交易类型	mhtOrderType	String(2)	Y	01 普通消费
商户订单币种类型	mhtCurrencyType	String(3)	Y	156 人民币
商户订单交易金额	mhtOrderAmt	String(1,22)	Y	单位(人民币): 分 整数, 无小数点
商户订单详情	mhtOrderDetail	String(1,1000)	Y	
商户订单超时时间	mhtOrderTimeOut	Number(4,0)	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
商户字符编码	mhtCharset	定值	Y	UTF-8
商户签名方法	mhtSignType	定值	Y	MD5
商户数据签名	mhtSignature	String(1,64)	Y	签名逻辑见接口附录说 见 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
```

## 附录 C

插件通知接口信息说明:

状态	提示信息
支付成功	[success]支付成功
支付失败	[fail]支付失败
支付取消	[cancel]支付取消