

现在支付  
聚合支付 SDK 对接文档  
安卓 V2.0.0

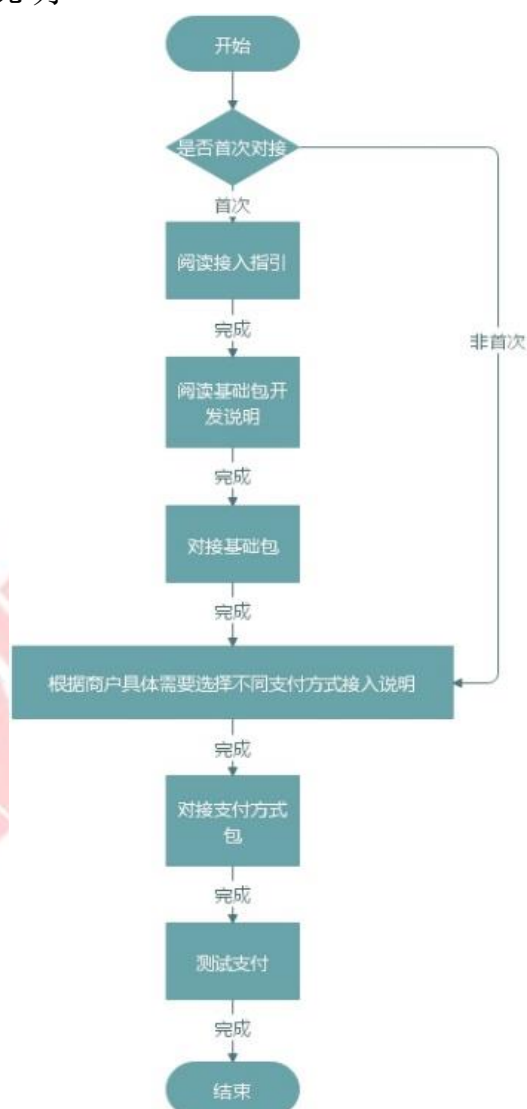
---

## 目录

一、	接入指引 .....	2
1.1	技术接入流程说明 .....	2
1.2	基本流程说明 .....	3
二、	基础 SDK 包对接说明 .....	3
2.1	导入资源: .....	3
2.2	配置 AndroidManifest.xml 文件 .....	3
2.3	编写构建脚本(仅 AndriodStudio, Eclipse 请跳过) .....	4
2.4	编写接入代码: .....	4
2.5	编辑混淆文件: .....	5
三、	银联支付 SDK 包对接说明 .....	8
3.1	导入资源: .....	8
3.2	配置 AndroidManifest.xml 文件 .....	8
3.3	编辑混淆文件: .....	8
四、	支付宝支付对接说明 .....	10
五、	微信原生 SDK 对接说明 .....	11
	接入注意: .....	11
六、	附录信息 .....	12

## 一、 接入指引

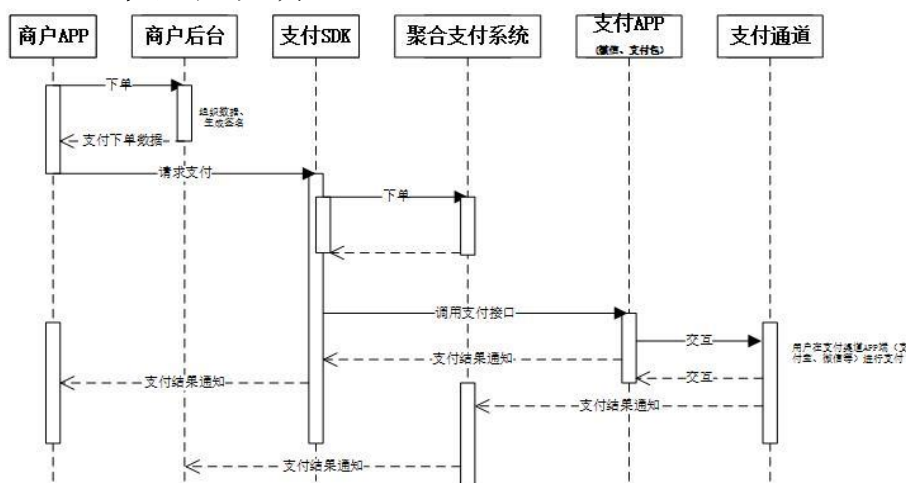
### 1.1 技术接入流程说明



资源文件说明:

1. 基础包:  
基础包用于为商户区分不同渠道之间的接入差异，为商户提供统一的对外接口。基础包包含 jar、so 文件。基础包是每个商户必须接入的部分。
2. 渠道包:  
渠道包用于为商户提供具体支付方式的服务。渠道包包含 jar 文件。渠道包商户根据自己的需求自行对接。

## 1.2 基本流程说明



1. 商户 APP 请求商户后台进行下单。
2. 商户后台根据聚合支付系统后台接口文档对订单数据、支付渠道信息进行组织和签名。
3. 商户 APP 接收到商户后台返回的数据，将数据传递给 SDK 的支付接口。
4. 支付 SDK 去聚合支付系统下单。
5. 支付 SDK 收到聚合支付系统返回的交易凭据，调起具体支付方式。
6. 消费者在支付 APP 上进行支付。
7. 支付 APP 将交易结果通知支付 SDK。
8. 支付 SDK 将结果通知给商户 APP。
9. 支付通道将交易结果通知给聚合支付系统。
10. 聚合支付系统将交易结果通知给商户后台。

## 二、基础 SDK 包对接说明

### 2.1 导入资源:

1. ipaynow\_base\_vxxx.jar:

拷贝到商户项目目录中 libs 目录下

2. libplugin\_phone.so:

拷贝到项目目录中 libs\<与目录同名的 cpu 架构名>目录下

### 2.2 配置 AndroidManifest.xml 文件

配置权限信息:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
```

配置 Activity 信息:

```
<activity
    android:name="com.ipaynow.plugin.presenter.PayMethodActivity"
    android:configChanges="keyboardHidden/navigation/orientation/screenSize"
    android:exported="false"
    android:theme="@android:style/Theme.Dialog" />
```

## 2.3 编写构建脚本(仅 AndriodStudio, Eclipse 请跳过)

在 app 模块的 build.gradle 添加资源集如下:

```
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.xiaohu"
        minSdkVersion 16
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

## 2.4 编写接入代码:

### 1. 实现支付结果监听器: ( 必须实现 )

用于监听支付结果以便商户根据结果进行展示

```
public class MainActivity extends AppCompatActivity implements ReceivePayResult
```

ReceivePayResult 接口:

方法: void onIpaynowTransResult(ResponseParams responseParams)

参数:

ResponseParams.respCode//结果响应码

ResponseParams.errorCode//结果错误码

ResponseParams.respMsg//结果响应信息

```
@Override
public void onIpaynowTransResult(ResponseParams arg0) {
    String respCode = arg0.respCode;
    String errorCode = arg0.errorCode;
    String errorMsg = arg0.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:" + errorMsg);
    } else if (respCode.equals("03")) {
        temp.append("交易状态:未知").append("\n").append("原因:" + errorMsg);
    } else {
        temp.append("respCode=").append(respCode).append("\n").append("respMsg=").append(errorMsg);
    }
    Toast.makeText(this, temp.toString(), Toast.LENGTH_LONG).show();
}
```

## 2. 插件初始化接口: ( 必须实现 )

用于 SDK 插件一些初始化操作，在 Activity 每次初始化时调用。

方法说明:

```
lpaynowPlugin mlpaynowplugin = lpaynowPlugin.getInstance().init(ActivityContext context);
```

## 3. 获得自带进度条: ( 选择实现 )

用于获取插件样式的进度条对象，用于商户走支付流程时展示与插件一致样式的进度条。

```
lpaynowPlugin mLoadingDialog=mlpaynowplugin.getDefaultLoading();
```

```
mLoadingDialog.setLoadingMsg("正在生成订单");
```

```
mLoadingDialog.show();
```

## 4. 设置结果监听器: ( 必须实现 )

用于设置商户实现的监听器，用于 SDK 将支付结果通知给商户

```
setCallResultReceiver(ReceivePayResult listener)
```

## 5. 交易方法: ( 必须实现 )

用于接收支付请求数据，以便为商户提供支付能力

```
pay(String requestMessage);
```

requestMessage 是表单组成的进行过数字签名的一组字符串，该请求串由商户后台生成，需要由商户后台，所以该请求串应由商户后台提供，具体请看后台说明文档。

## 6. 小程序支付环境: ( 选择实现 )

用于设置小程序支付时调用的微信小程序环境。(0:生产, 1:测试, 2:预览)

```
setMiniProgramEnv(int envFlag)
```

## 7. 释放资源方法: ( 必须实现 )

用于释放一些引用，防止内存泄露，推荐在界面销毁时调用。

```
mlpaynowplugin.onActivityDestroy();
```

## 2.5 编辑混淆文件:

打包应用如需代码混淆，需要保留与支付 SDK 相关的方法不被混淆，请加入以下配置，否则会出错:

可参考 resource/proguard-rules.pro 文件

```
-optimizationpasses 5
-dontusemixedcaseclassnames
-dontskipnonpubliclibraryclasses
-dontskipnonpubliclibraryclassmembers
-dontpreverify
-verbose
-printmapping proguardMapping.txt
-optimizations !code/simplification/cast,!field/*,!class/merging/*
-keepattributes *Annotation*,InnerClasses
-keepattributes Signature
-keepattributes SourceFile,LineNumberTable
-ignorewarnings

#-----

#-----默认保留区-----

-keep public class * extends android.app.Activity
-keep public class * extends android.app.Application
-keep public class * extends android.app.Service
```



---

```

-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keep public class * extends android.app.backup.BackupAgentHelper
-keep public class * extends android.preference.Preference
-keep public class * extends android.view.View
-keep public class com.android.vending.licensing.ILicensingService
-keep class android.support.** {*; }

-keepattributes *Annotation*
-keepattributes *JavascriptInterface*
-keepclassmembers class * { @android.webkit.JavascriptInterface <methods>; }

-keepclasseswithmembers class * {
    native <methods>;
}

-keepclassmembers class * extends android.app.Activity{
    public void *(android.view.View);
}

-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

-keep public class * extends android.view.View{
    *** get*();
    void set*(***);
    public <init>(android.content.Context);
    public <init>(android.content.Context, android.util.AttributeSet);
    public <init>(android.content.Context, android.util.AttributeSet, int);
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet);
    public <init>(android.content.Context, android.util.AttributeSet, int);
}

-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;
}

-keepclassmembers class * implements java.io.Serializable {
    static final long serialVersionUID;
    private static final java.io.ObjectStreamField[] serialPersistentFields;
    private void writeObject(java.io.ObjectOutputStream);
    private void readObject(java.io.ObjectInputStream);
    java.lang.Object writeReplace();
    java.lang.Object readResolve();
}

-keep class **.R$* {
    *;
}

```

---

# 删除代码中 Log 相关的代码

```
-assumenosideeffects class cn.ipaynow.webankwallet.plugin.log.LogUtils {  
    public static *** v(...);  
    public static *** i(...);  
    public static *** w(...);  
    public static *** d(...);  
    public static *** e(...);  
}
```

#-----项目专有配置区-----

```
-keep class com.alipay.android.app.IAlixPay{*};  
-keep class com.alipay.android.app.IAlixPay$Stub{*};  
-keep class com.alipay.android.app.IRemoteServiceCallback{*};  
-keep class com.alipay.android.app.IRemoteServiceCallback$Stub{*};  
-keep class com.alipay.sdk.app.PayTask{ public *};  
-keep class com.alipay.sdk.auth.AlipaySDK{ public *};  
-keep class com.alipay.sdk.auth.APAuthInfo{ public *};  
-keep class com.alipay.mobilesecuritysdk.*  
-keep class com.ut.*  
-keep class cn.gov.pbc.tsm.*{*};  
-keep class com.UCMobile.PayPlugin.*{*};  
-keep class com.unionpay.*{*};  
-dontwarn com.unionpay.**  
  
-keep class com.ipaynow.plugin.api.IpaynowPlugin{  
    <fields>;  
    <methods>;  
}  
-keep class com.ipaynow.plugin.manager.route.dto.RequestParams{  
    <fields>;  
    <methods>;  
}  
-keep class com.ipaynow.plugin.manager.route.dto.ResponseParams{  
    <fields>;  
    <methods>;  
}  
-keep class com.ipaynow.plugin.manager.route.impl.ReceivePayResult{  
    <fields>;  
    <methods>;  
}  
-keep class com.alipay.android.app.IAlixPay {  
    <fields>;  
    <methods>;  
}  
-keep class com.ipaynow.plugin.utils.StringUtils{  
    <fields>;  
    <methods>;  
}  
-keep class com.alipay.android.app.IRemoteServiceCallback {  
    <fields>;  
    <methods>;  
}
```



```
-keep class com.ipaynow.plugin.utils.NativeUtils{
    <fields>;
    <methods>;
}
```

### 三、 银联支付 SDK 包对接说明

#### 3.1 导入资源:

1. UPPayAssistEx.jar (银联技术包)  
拷贝到商户项目目录中 libs 目录下
2. UPPayPluginExPro.jar (银联技术包)  
拷贝到商户项目目录中 libs 目录下
3. data.bin(银联插件资源压缩包)  
拷贝到商户项目目录中的 assets 目录下
4. libentryexpro.so (银联插件动态库文件)  
拷贝到项目目录中 libs\<与目录同名的 cpu 架构名>目录下
5. libuptsmaddon.so(三星 pay、华为 pay 使用的动态库)  
拷贝到项目目录中 libs\<与目录同名的 cpu 架构名>目录下
6. libuptsmaddonmi.so  
拷贝到项目目录中 libs\<与目录同名的 cpu 架构名>目录下

#### 3.2 配置 AndroidManifest.xml 文件

##### 1. 配置权限信息:

```
1. <uses-permission android:name="android.permission.INTERNET"/>
2. <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
3. <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
4. <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
5. <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
6. <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
7. <uses-permission android:name="android.permission.NFC" />
8. <uses-feature android:name="android.hardware.nfc.hce"/>
9. <uses-permission android:name="android.permission.RECORD_AUDIO"/>
10. <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
11. <uses-permission android:name="org.simalliance.openmobileapi.SMARTCARD" />
```

##### 2. 配置 Activity 信息:

```
<activity
    android:name="com.unionpay.uppay.PayActivity"
    android:configChanges="orientation|keyboardHidden"
    android:excludeFromRecents="true"
    android:label="现在支付SDK演示样例"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="adjustResize" />
```

注意：版本编译时的 target 应选用 Open Mobile API，建议 level19 以上。

#### 3.3 编辑混淆文件:

```
-keep public class com.unionpay.uppay.net.HttpConnection {*; }
-keep public class com.unionpay.uppay.net.HttpParameters {*; }
-keep public class com.unionpay.uppay.model.BankCardInfo {*; }
-keep public class com.unionpay.uppay.model.PAAInfo {*; }
-keep public class com.unionpay.uppay.model.ResponseInfo {*; }
-keep public class com.unionpay.uppay.model.PurchaseInfo {*; }
-keep public class com.unionpay.uppay.util.DeviceInfo {*; }
```

---

```
-keep public class com.unionpay.uppay.util.PayEngine {*;}  
-keep public class com.unionpay.utils.UPUtils {*;}  
-dontwarn com.unionpay.**
```

---

## 四、 支付宝支付对接说明

根据官方指导说明对接支付宝官方支付 SDK:

客户端 Android 集成流程 - 支付宝文档中心 ([alipay.com](https://alipay.com))

## 五、 微信原生 SDK 对接说明

根据官方指导说明对接微信官方支付 SDK:

[开发指引-APP 支付 | 微信支付商户平台文档中心 \(qq.com\)](#)

接入注意:

1. 该支付方式由于微信校验您应用的包名签名，请务必保持您应用的包名签名与微信平台上  
面登记的包名签名一致，否则无法调起。

applicationId 与微信平台上面一致:

```
android {  
    compileSdkVersion 28  
    defaultConfig {  
        applicationId "com.kzjx.kzjx"  
        minSdkVersion 16  
        targetSdkVersion 28  
        versionCode 1  
        versionName "1.0"  
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
    }  
}
```

应用包名与微信平台上面一致:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.kzjx.kzjx">
```

## 六、 附录信息

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	失败时会返回
响应信息	respMsg	N	未知和失败时会返回

插件错误码说明:

字段名称	错误信息说明
PE001	插件不支持该渠道交易
PE002	网络连接超时
PE003	现在支付获得交易号失败
PE004	渠道方交易失败
PE005	支付插件其他未知错误
PE006	渠道方不支持此种支付方式
PE007	未安装渠道方支付用客户端
PE008	支付用渠道客户端版本过低
PE009	渠道交易结果未知
PE010	渠道方其他未知失败
PE011	现在支付接口其他失败情况
PE012	商户未添加渠道子包
PE013	商户未添加视图控件子包
PE015	小程序支付错误