金融平台SDK集成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 版本更改说明 | 更改日期 | 更改人 | 发布时间 |
| V1.0.0 | 1.创建  2.添加iOS集成步骤 | 2017-3-2 | 李勇 |  |
| V1.0.1 | 1.添加安卓集成步骤 2.修改iOS集成步骤 | 2017-3-7 | 李庆军  李勇 |  |
|  |  |  |  |  |

目录

[一、概述 1](#_Toc476732470)

[二、SDK集成 1](#_Toc476732471)

[1、iOS项目集成SDK 1](#_Toc476732472)

[1.1、引入文件 1](#_Toc476732473)

[1.1.1、引入头文件 1](#_Toc476732474)

[1.1.2、引入库文件 1](#_Toc476732475)

[1.2、项目配置 2](#_Toc476732476)

[1.2.1、 Other Link Flags 2](#_Toc476732477)

[1.2.2、添加支持库 2](#_Toc476732478)

[1.2.3、Header Search Path 2](#_Toc476732479)

[1.2.4、Keychain Sharing 3](#_Toc476732480)

[1.2.5、bitcode 3](#_Toc476732481)

[1.3、支付操作 4](#_Toc476732482)

[1.3.1、注册appID 4](#_Toc476732483)

[1.3.2、设置图片位置 4](#_Toc476732484)

[1.3.3、调起支付 4](#_Toc476732485)

[2、Android项目集成SDK 7](#_Toc476732486)

[2.1、引入aar文件 7](#_Toc476732487)

[2.2、修改build.gradle配置文件 7](#_Toc476732488)

[2.3、重新编译项目 8](#_Toc476732489)

[2.4、修改Manifest配置文件 8](#_Toc476732490)

[2.5、定制一个本应用程序 Application 类 9](#_Toc476732491)

[2.6、添加混淆规则 10](#_Toc476732492)

[2.7、支付接口调用 10](#_Toc476732493)

[2.8、支付结果获取和处理 13](#_Toc476732494)

# 一、概述

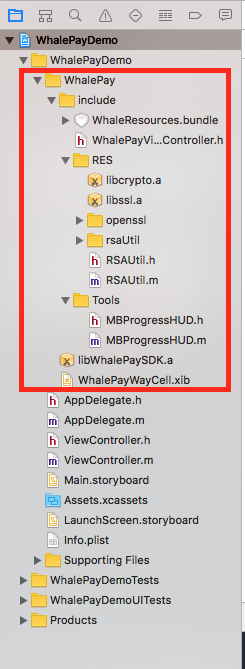
# 二、SDK集成

# 1、iOS项目集成SDK

## 1.1、引入文件

### 1.1.1、引入头文件

在whalePay文件夹中，CER文件夹为加密解密相关代码，tools里面为sdk需要工具类，接入者对文件内方法进行修改可能会导致错误，如有需要可添加方法。WhaleResources.bundle为资源文件，接入者可替换内部图片，要保持名称不变。



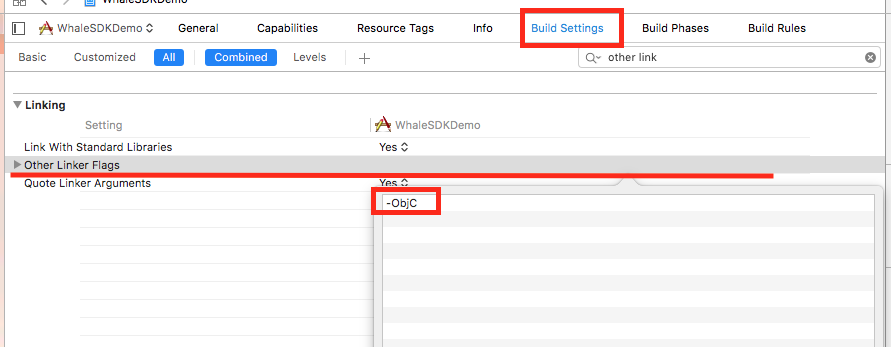
### 1.1.2、引入库文件

在项目中引入libWhalePaySDK.a

## 1.2、项目配置

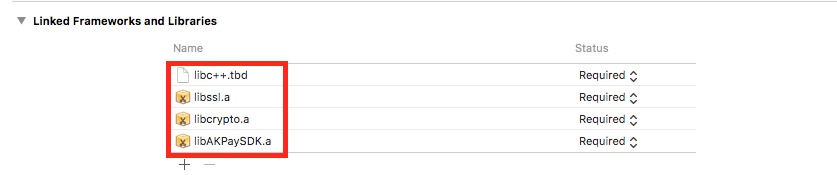
### 1.2.1、 Other Link Flags

在targets->build Setting 中配置Other Link Flags 中的参数设定为-ObjC



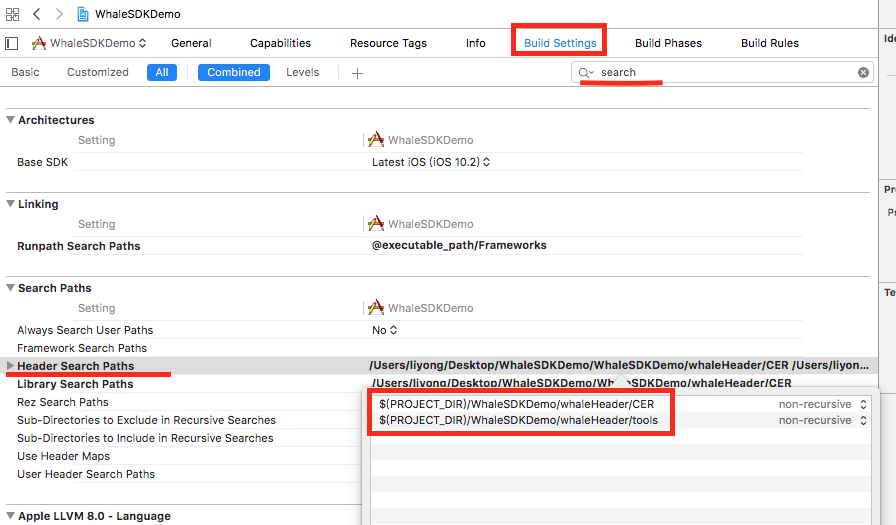
### 1.2.2、添加支持库

在targets->General中配置libc++.tbd文件，将libcrypto.a和libssl.a两个类库拖进工程中，这两个属于openssl库的静态库



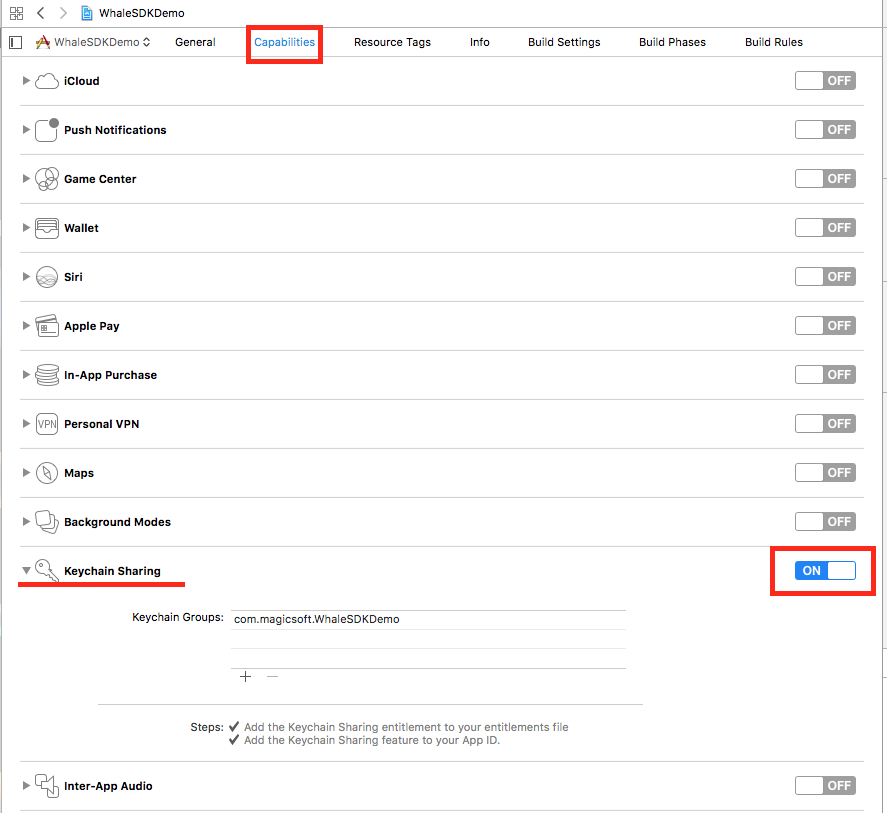
### 1.2.3、Header Search Path

将Demo中的whaleHeader文件夹拖进工程，如果出现头文件找不到的问题，请在targets->build Setting 中配置Header Search Path 配置CER文件夹的路径



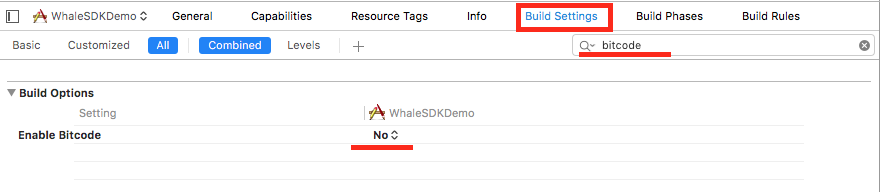
### 1.2.4、Keychain Sharing

xcode8更新后，插件加密方法完全失效，需要打开targets -> Capabilities -> Keychain Sharing



### 1.2.5、bitcode

真机运行时如果出现bitcode导致的错误ld:'/Users/liyong/Desktop/whaleSDK/WhalePayTest/WhalePayDemo/WhalePay/libWhalePaySDK.a(bio\_lib.o)' does not contain bitcode. You must rebuild it with bitcode enabled (Xcode setting ENABLE\_BITCODE), obtain an updated library from the vendor, or disable bitcode for this target. for architecture arm64可将值设置为no



## 1.3、支付操作

### 1.3.1、注册appID

使用支付工具前需要在AppDelegate.m的 - (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions;方法中向金融平台注册appid，参数如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 备注 |
| appkey | string | 金融平台分发 |  |
| appsercet | string | 金融平台分发 |  |
| wxAppid | string | 微信申请到的appid |  |
| aliSchemel | string | 支付宝schemel |  |

代码如下：

//向金融平台注册appID

/\*\*

\* 打开app时注册 包含：1.注册APPID（微信）2.appkey 3.appsecret

\* key value

\* aliSchemel 支付宝schemel

\* wxAppid 微信支付ID，例如“wx42967af88ec67866”

\* appkey 接入应用新增成功后，系统将自动生成20位的appkey和32位的appsercet

\* appsecret 接入应用新增成功后，系统将自动生成20位的appkey和32位的appsercet

\*/

[[AKPayViewController sharedInstance] setAppId:@{

@"appkey" : @"",

@"appsecret" : @" "

}];

### 1.3.2、设置图片位置

如果需要的话可以在调起支付前通过[AKPayViewController sharedInstance] .bundleName = @"";更改库文件加载图片的位置，要保证更改的图片名称不变。----暂不可用！

### 1.3.3、调起支付

调起支付时需要首先配置好订单信息，下面只填写了必要参数，未填写可选参数。

必要参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 备注 |
| pno | string | 平台账号 | 本app相对于金融平台对应的ID号 |
| cno | string | 用户针对金融平台中当前应用的身份 |  |
| payTypeArray | array | 表示“app支持的支付方式”，在数组里面包含number类型的数字，表示支付通道的atid |  |
| receiptArray | array | 表示“支持的收款通道”，在数组里面包含带有以下属性的实体类型  atid;//收款通道类型  name;//收款通道名称  ano;//收款帐号编号  money;//此收款通道应收金额  info;//附加信息  isDefault;//是否为默认收款账号  ctype;//通道类型 1:第三方通道 4:虚拟货币通道  type;//饭票类型 fanpiao：全国饭票 lfanpiao：地方饭票 |  |
| money | string | 交易金额 |  |
| orderno | string | 接入方生成的订单编号 |  |
| content | string | 交易说明（显示给用户的内容） |  |
| userId | string | 本app里面当前用户对应的ID号 |  |

可选参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 备注 |
| orgsubaccount | string | 支付账号子账号修改信息 |  |
| destsubaccount | string | 收款账号子账号修改信息 |  |
| fixedorgmoney | string | 支付金额固定，0不固定，1固定 |  |
| username | string | 本app里面当前用户对应的用户名 |  |
| detail | string | 商品描述 |  |
| mobile | string | 本app里面当前用户的手机号 |  |
| province | string | 省 |  |
| city | string | 市 |  |
| block | string | 行政区 |  |
| gbName | string | 小区名称 |  |
| remoteid | string | 调用方ip地址 | 当使用微信支付时必须传递当前app的ip地址 |
| callback | string | 业务系统回调 |  |
| stoptime | string | 交易失效时间 |  |
| starttime | string | 交易生效时间 |  |

WPOrder \*wpOrder = [[WPOrder alloc] init];

wpOrder.pno = @"123456789";

wpOrder.cno = @"987654321";

wpOrder.orderno = @"123123123";

wpOrder.content = @"机械键盘";

wpOrder.userId = @"233";

wpOrder.remoteip = @"192.0.1.9";

wpOrder.fixedorgmoney = @"";

wpOrder.payTypeArray = @[@(1),@(2),@(4),@(31),@(22),@(24)];

WPReceiptType \*receipt1 = [[WPReceiptType alloc] init];

receipt1.atid = @"31";

receipt1.name = @"彩之云邻花钱";

receipt1.ano = @"1031c5a4efa552b24491bda9d212abbb";

receipt1.money = @"100";

receipt1.info = @"不打折";

receipt1.isDefault = @"";

receipt1.ctype = @"1";

receipt1.type = @"lhq";

WPReceiptType \*receipt2 = [[WPReceiptType alloc] init];

receipt2.atid = @"4";

receipt2.name = @"微信";

receipt2.ano = @"1004b4aac2537a3b40cdaf3bac370ad0";

receipt2.money = @"110";

receipt2.info = @"微信不打折";

receipt2.isDefault = @"";

receipt2.ctype = @"1";

receipt2.type = @"weixin";

WPReceiptType \*receipt3 = [[WPReceiptType alloc] init];

receipt3.atid = @"2";

receipt3.name = @"彩之云全国饭票";

receipt3.ano = @"eb65c3ca8016462e9dc4d9b1a276a8d";

receipt3.money = @"0.01";

receipt3.info = @"饭票打一折";

receipt3.isDefault = @"";

receipt3.ctype = @"4";

receipt3.type = @"fanpiao";

WPReceiptType \*receipt4 = [[WPReceiptType alloc] init];

receipt4.atid = @"1";

receipt4.name = @"北苑地方饭票";

receipt4.ano = @"eb65c3ca8016462ebdc4d9b1a276a8d";

receipt4.money = @"0.41";

receipt4.info = @"饭票打一折";

receipt4.isDefault = @"";

receipt4.ctype = @"4";

receipt4.type = @"lfanpiao";

wpOrder.receiptArray = @[receipt1,receipt2,receipt3,receipt4];

WhalePayViewController \*whalePayVC = [WhalePayViewController sharedInstance];

whalePayVC.bundleName = @"";

[whalePayVC createPayment:wpOrder viewController:self withCompletion:^(NSDictionary \*result) {

if ([[result objectForKey:@"code"] integerValue] == 0){

[MBProgressHUD showMessage:[result objectForKey:@"message"] toView:self.view];

}else{

[MBProgressHUD showMessage:[result objectForKey:@"message"] toView:self.view];

NSLog(@"%@",[result objectForKey:@"message"]);

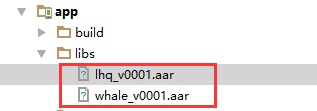
}

}];

# 2、Android项目集成SDK

## 2.1、引入aar文件

将whale\_vxxxx.aar文件和lhq\_vxxxx.aar文件放入商户应用工程的libs目录下,如果工程下面没有libs文件夹，则自己新建一个，如下图。



PS: whale\_vxxxx.aar中的xxxx代表交付aar文件时的当前版本号，例如：whale\_v0001.aar

## 2.2、修改build.gradle配置文件

如下添加：

repositories {

flatDir {

dirs 'libs' }

}

这个是添加一个本地仓库，并把libs目录作为仓库的地址。

然后在dependencies中添加下面一行代码：

compile(name:'whale­\_vxxxx', ext:'aar')

compile(name:'lhq­\_vxxxx', ext:'aar')

PS:这里的aar名一定要与接入的arr包名相同

其中name就是libs目录下whale\_v0001.aar文件名称whale\_v0001，

ext，就是whale\_v0001.aar的扩展名aar

完整添加如下：

repositories {

flatDir {

dirs 'libs'

}

}

dependencies {

compile(name: 'whale\_v0001', ext: 'aar')

compile(name: 'lhq\_v0001', ext: 'aar')

}

## 2.3、重新编译项目

从工具栏依次选择："Build"-->"rebuild project"

## 2.4、修改Manifest配置文件

在应用工程的AndroidManifest.xml文件里面添加声明，示例代码如下：

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="您应用的包名">

<!—权限声明 -->

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"/>

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>

<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/>

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

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"/>

<application

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:name="Your Application Name">

<!—支付界面 -->

<activity android:name="com.magicsoft.whale.activity.PayActivity"></activity>

<!-- 插件控制逻辑处理界面 -->

<activity android:name="com.lhqpay.ewallet.activity.LHQActivity"

></activity>

<!-- 插件控制逻辑处理界面 -->

<activity android:name="com.lhqpay.ewallet.activity.CSHActivity"

></activity>

<!-- 插件选择银行卡支付界面 -->

<activity android:name="com.lhqpay.ewallet.activity.CardActivity"

android:windowSoftInputMode="stateHidden|adjustPan" ></activity>

<activity android:name="com.lhqpay.ewallet.activity.CSHCardActivity"

android:windowSoftInputMode="stateHidden|adjustPan" ></activity>

<!-- 插件银行卡列表界面 -->

<activity android:name="com.lhqpay.ewallet.activity.CardListActivity"></activity>

<activity android:name="com.lhqpay.ewallet.activity.CSHCardListActivity"></activity>

<!-- 插件填写银行卡信息支付界面 -->

<activity android:name="com.lhqpay.ewallet.activity.BillActivity"

android:windowSoftInputMode="stateHidden|adjustPan"></activity>

<activity android:name="com.lhqpay.ewallet.activity.CSHBillActivity"

android:windowSoftInputMode="stateHidden|adjustPan" ></activity>

<!-- 支持的银行卡界面 -->

<activity android:name="com.lhqpay.ewallet.activity.SupportBankListActivity"

android:windowSoftInputMode="stateHidden|adjustPan"></activity>

<meta-data

android:name="WHALE\_APPKEY"

android:value="您应用的AppKey"></meta-data>

<meta-data

android:name="WHALE\_APPSECRET"

android:value="您应用的AppSecret"></meta-data>

</application>

</manifest>

## 2.5、定制一个本应用程序 Application 类

需要在 AndoridManifest.xml 里配置。请参考上面 AndroidManifest.xml 片断，

在onCreate()方法里加入代码

ImageLoaderUtil.initImageLoader(getApplicationContext());

示例代码如下：

public class ExampleApplication extends Application {

@Override

public void onCreate() {

super.onCreate();

ImageLoaderUtil.initImageLoader(getApplicationContext());

}

}

## 2.6、添加混淆规则

在商户应用工程的proguard-project.txt里添加以下相关规则：

-keep class com.magicsoft.whale.\*\*{ \*; }

-keep class com.magicsoft.whale.adapter.\*\* { \*; }

-keep class com.magicsoft.whale.entity.\*\* { \*; }

-keep class com.magicsoft.whale.util.\*\* { \*; }

-keep public class com.magicsoft.whale.R$\*{

public static final int \*;

}

-keep class com.lhqpay.ewallet.\*\*{ \*; }

-keep class com.lhqpay.ewallet.adapter.\*\* { \*; }

-keep class com.lhqpay.ewallet.basil.\*\* { \*; }

-keep class com.lhqpay.ewallet.entity.\*\* { \*; }

-keep class com.lhqpay.ewallet.utils.\*\* { \*; }

-keep class com.lhqpay.ewallet.wedgit.\*\* { \*; }

-keep class com.lhqpay.ewallet.lhqpay.listener.\*\* { \*; }

-keep public class com.lhqpay.ewallet.R$\*{

public static final int \*;

}

## 2.7、支付接口调用

在调用支付接口的时候，最好新启一个线程，然后创建一个支付实例pay，设置pay参数后调用WhalePay akPay = WhalePay.getInstance(this);最后调用whalePay.pay(pay);方法并传入pay实例，

pay实体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 含义 | 备注 |
| pno | string | 平台账号 | 本app相对于金融平台对应的ID号  （必填） |
| cno | string | 用户针对金融平台中当前应用的身份 | （必填） |
| payTypeArray | List< AppPayChannel > | app支持的支付通道  AppPayChannel实体类型包含参数字段：  atid;//通道类型  name;//通道名称 | （必填） |
| receiptArray | List< PayChannel > | 商户支持的收款通道  PayChannel实体类型包含参数字段：  atid;//收款通道类型  name;//收款通道名称  ano;//收款帐号编号  money;//此收款通道应收金额  info;//附加信息  isDefault;//是否为默认收款账号  ctype;//通道类型 1:第三方通道 4:虚拟货币通道  type;//饭票类型 fanpiao：全国饭票 lfanpiao：地方饭票 | （必填） |
| money | double | 交易金额 | （必填） |
| orderno | string | 接入方生成的订单编号 | （必填） |
| content | string | 交易说明（显示给用户的内容） | （必填） |
| userId | string | 本app里面当前用户对应的ID号 | （必填） |
| orgsubaccount | string | 支付账号子账号修改信息（保留字段） | （可选） |
| destsubaccount | string | 收款账号子账号修改信息（保留字段） | （可选） |
| fixedorgmoney | string | 支付金额固定，0不固定，1固定 | （可选） |
| username | string | 本app里面当前用户对应的用户名 | （可选） |
| detail | string | 商品描述 | （可选） |
| mobile | string | 本app里面当前用户的手机号 | （可选） |
| province | string | 省 | （可选） |
| city | string | 市 | （可选） |
| block | string | 行政区 | （可选） |
| gbName | string | 小区名称 | （可选） |
| remoteid | string | 调用方ip地址 | （可选） |
| callback | string | 业务系统回调 | （可选） |
| stoptime | string | 交易失效时间 | （可选） |
| starttime | string | 交易生效时间 | （可选） |

代码示例：

PayResp pay = new PayResp();

//接入平台账号 必填

pay.setPno("123456789");

//交易金额 必填

pay.setCno("987654321");

//交易金额 必填

pay.setMoney(amount);

//用户编号 必填

pay.setUserNo("111");

//用户姓名 选填

pay.setUserName("测试用户");

//订单类型 选填

pay.setOrdertype("0");

//业务系统交易编号 必填

pay.setOrderno("20161008143504" + BaseUtil.createRandomNumber(1000, 9000));

//交易说明（显示给用户的内容） 必填

pay.setContent("商品");

// app支持的支付通道

pay.setReceiptArray(receiptArray);

//商户支持的收款通道

pay.setPayTypeArray(payTypeArray);

//支付金额固定，0不固定，1固定

pay.setFixedorgmoney("0");

//交易描述 选填

pay.setDetail("商品描述");

//手机号 选填

pay.setMobile("12345678901");

//支付账号类型，微信支付宝双乾交易时为0

pay.setOrgtype(0);

//支付帐号编号，微信支付宝双乾交易时为0

pay.setOrgaccountno("0");

// 交易生效时间

pay.setStarttime("");

// 交易失效时间

pay.setStoptime("");

//业务系统回调地址

pay.setCallback("");

//调用方ip地址

pay.setRemoteip("");

/\* 小区信息 均为option\*/

//省

pay.setProvince("");

//城市

pay.setCity("");

//区

pay.setBlock("");

//小区名称

pay.setGbName("");

WhalePay whalePay = WhalePay.getInstance(this);

whalePay.pay(pay);

## 2.8、支付结果获取和处理

本SDK插件采用intent进行跳转传值与返回结果，商户应用客户端通过

onActivityResult来接收来自SDK的反馈信息，其中resultCode为10010，实例代码如下：

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (10010 == resultCode) {

// WhalePay返回

if (data != null) {

StringBuilder sb = new StringBuilder();

int code = data.getIntExtra("code", -1);

if (code == 10088) {

//支付成功

Toast.makeText(this, sb.toString(), Toast.LENGTH\_SHORT)

.show();

} else {

Toast.makeText(this, data.getStringExtra("message"),

Toast.LENGTH\_SHORT).show();

}

}

}

}