

# Android支付Sdk接入指引

## 文档修订记录

版本号	时间	修订人	修订原因
V1.1.0	2021-04-28	庄纪光	优化文档

## 集成步骤

### 配置ndk

在app module的build.gradle下配置,

```
// 按需配置ndk, 设置要保留的so库, 比如:  
// 一般 只保留 armeabi-v7a 或者 armeabi 其中一个  
ndk {  
    abiFilters "armeabi-v7a", "armeabi", "x86"  
}
```

### 引入依赖

```
// 引入支付aar  
implementation (name: 'BusinessPay', ext: 'aar')  
// 注意: 以下依赖如果宿主工程中已引入, 可忽略。  
implementation 'com.tencent.mm.opensdk:wechat-sdk-android-without-mta:5.5.8'  
implementation 'com.google.code.gson:gson:2.8.2'  
implementation 'com.squareup.okhttp3:okhttp:3.11.0'  
implementation 'com.squareup.retrofit2:retrofit:2.4.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.4.0'  
implementation 'com.squareup.retrofit2:adapter-rxjava2:2.4.0'  
implementation 'org.bouncycastle:bcprov-jdk16:1.46'  
implementation 'com.android.support:recyclerview-v7:27.1.1'
```

### 微信支付特别注意

1. applicationId
2. 包名.wxapi.WXPayEntryActivity
3. 应用的签名

## 添加微信支付回调类

了解更多信息，可参考[微信开发平台接入指引](#)。

```
package 你应用的包名.wxapi;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import com.pasc.business.ewallet.R;
import com.pasc.business.ewallet.business.StatusTable;
import com.pasc.business.ewallet.openapi.PASCPay;
import com.pasc.business.ewallet.result.PayResult;
import com.tencent.mm.opensdk.constants.ConstantsAPI;
import com.tencent.mm.opensdk.modelbase.BaseReq;
import com.tencent.mm.opensdk.modelbase.BaseResp;
import com.tencent.mm.openapi.IWXAPI;
import com.tencent.mm.openapi.IWXAPIEventHandler;
import com.tencent.mm.openapi.WXAPIFactory;

public class WXPAYEntryActivity extends Activity implements IWXAPIEventHandler
{
    private IWXAPI api;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ewallet_wechatpay_result);

        api = WXAPIFactory.createWXAPI(this,
PASCPay.getInstance().getWechatPayAppID());
        api.handleIntent(getIntent(), this);
    }

    @Override
    protected void onNewIntent(Intent intent) {
        super.onNewIntent(intent);
        setIntent(intent);
        api.handleIntent(intent, this);
    }

    @Override
    public void onReq(BaseReq req) {
    }

    @Override
    public void onResp(BaseResp resp) {
        if(resp.getType()==ConstantsAPI.COMMAND_PAY_BY_WX){
            if(resp.errCode==BaseResp.ErrCode.ERR_OK){
```

```

        PASC Pay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_SUCCESS, "");
    }else if (resp.errCode==BaseResp.ErrCode.ERR_USER_CANCEL){
        PASC Pay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_CANCEL, "");
    }else if (resp.errCode==BaseResp.ErrCode.ERR_SENT_FAILED){
        PASC Pay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_FAILED, resp.errStr);
    }else {
        PASC Pay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_WAITING, resp.errStr);
    }
    finish();
}
}
}

```

## 添加微信小程序支付回调类

了解更多信息，可参考[微信开放平台接入指引](#)。

```

package 你应用的包名.wxapi;
import android.app.Activity;
import android.os.Bundle;
import android.support.annotation.Nullable;
import com.chinaums.pppay.unify.UnifyPayPlugin;
import com.pasc.business.ewallet.business.StatusTable;
import com.pasc.business.ewallet.common.utils.LogUtil;
import com.pasc.business.ewallet.openapi.PASC Pay;
import com.pasc.business.ewallet.result.PayResult;
import com.tencent.mm.opensdk.constants.ConstantsAPI;
import com.tencent.mm.opensdk.modelbase.BaseReq;
import com.tencent.mm.opensdk.modelbase.BaseResp;
import com.tencent.mm.opensdk.modelbiz.WXLaunchMiniProgram;
import com.tencent.mm.opensdk.openapi.IWXAPI;
import com.tencent.mm.opensdk.openapi.IWXAPIEventHandler;
import com.tencent.mm.opensdk.openapi.WXAPIFactory;

public class WXEntryActivity extends Activity implements IWXAPIEventHandler {
    private static final String TAG = "WXEntryActivity";
    private IWXAPI api;
    @Override protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        api = WXAPIFactory.createWXAPI(this,
PASC Pay.getInstance().getWechatPayAppID());
        //注意：第三方开发者如果使用透明界面来实现WXEntryActivity，需要判断handleIntent的返回值，如果返回值为false，则说明入参不合法未被SDK处理，应finish当前透明界面，避免外部通过传递非法参数的Intent导致停留在透明界面，引起用户的疑惑
        try {

```

```

        if (!api.handleIntent(getIntent(), this)) {
            finish();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override public void onReq(BaseReq baseReq) {
    LogUtil.logd(TAG, "onReq: " + baseReq.toString());
}

@Override public void onResp(BaseResp baseResp) {
    LogUtil.logd(TAG, "onResp-> errCode: " + baseResp.errCode + ",errStr: " +
baseResp.errStr);
    if (baseResp.getType() == ConstantsAPI.COMMAND_LAUNCH_WX_MINIPROGRAM) {
        WXLaunchMiniProgram.Resp launchMiniProResp = (WXLaunchMiniProgram.Resp)
baseResp;
        String extraData = launchMiniProResp.extMsg; // 对应下面小程序中的app-
parameter字段的value
        finish();
        if (baseResp.errCode == BaseResp.ErrCode.ERR_OK) {
            PASCPay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_SUCCESS, "");
        } else if (baseResp.errCode == BaseResp.ErrCode.ERR_USER_CANCEL) {
            PASCPay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_CANCEL, "");
        } else if (baseResp.errCode == BaseResp.ErrCode.ERR_SENT_FAILED) {
            PASCPay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_FAILED, baseResp.errStr);
        } else {
            PASCPay.getInstance().notifyPayResult(StatusTable.PayType.WECHAT,
PayResult.PAY_WAITING, baseResp.errStr);
        }
        UnifyPayPlugin.getInstance(this).getWXListener().onResponse(this,
baseResp);
    }
}
}

```

注意：将WXPayEntryActivity和WXEntryActivity添加到宿主工程的AndroidManifest.xml文件中。

```
<activity
    android:name="${applicationId}.wxapi.WXPayEntryActivity"
    android:screenOrientation="portrait"
    android:exported="true" />
<activity
    android:name="${applicationId}.wxapi.WXEntryActivity"
    android:exported="true"
    android:launchMode="singleTop"
    android:theme="@android:style/Theme.Translucent" />
```

## Sdk Api接口

### 配置信息

请使用提供的AppId、加盐密钥替换、公钥、微信支付AppId：

1. 接入AppId(AppId): 由支付平台提供
2. Sdk接入密钥(secretKey): 由支付平台提供
3. Sdk接入公钥 (publicKey) : 由支付平台提供
4. Sdk接入微信支付AppId (wxPayAppId) : 客户端自己申请
5. Sdk接入域名和前缀 HOST\_URL: 每个客户端会分配一个支付 (域名+前缀)
6. 支付宝支付申请
7. 请提前开通申请好微信和支付宝的 支付功能, 然后把微信的appid 和支付宝的公私钥 等商户信息发给 支付平台后台对接人

### 测试环境信息

AppId: eb9cd984905a5b1891a9bdb0a50a0e44

secretKey: b0ec98cd545f566a933b9ac781f1c00d

publicKey:

046F160C753F66D14EC40E136D2317EFAF027D64D022C68D1FDA2189F8F8AD95916514191F22B  
DB7B829B55F634634E6825C303B245E81661039F5DE6EFB96E932

### 支付Sdk初始化方式1

在App启动的时候, 在主工程的Application主进程里对支付Sdk进行初始化, Api接口如下:

```
PASCPay.getInstance().init(@NonNull Application application, @NonNull String
publicKey, @NonNull String AppId, @NonNull String secretKey, @NonNull String
wxPayAppId, boolean isDebug, String hostAndGateWay);
```

### 支付Sdk初始化方式2

```
PASCPay.getInstance().init (
    new PayBuilder.Build (application)
    .publicKey (publicKey)
    .appId (AppId)
    .secretKey (secretKey)
    .wxPayAppId (wxPayAppId)
    .isDebug (isDebug)
    .hostAndGateWay (HOST_URL)
    .build ());
```

参数	类型	是否必须	说明
application	Application	是	上下文环境
publicKey	String	是	Sdk接入公钥
AppId	String	是	Sdk接入AppId
secretKey	String	是	Sdk接入密钥
wxPayAppId	String	是	微信支付AppId
isDebug	boolean	是	是否打印log
hostAndGateWay	String	是	分配给客户端的支付（域名+前缀）

**重要提示：**正式发布生产环境前务必将isDebug设置为false。

## 进入钱包首页

进入钱包首页，会检查当前用户是否已开通钱包账户，接口如下：

```
PASCPay.getInstance().open(@NonNull Context context, @Nullable String
merchantNo, @NonNull String memberNo, @NonNull OnOpenListener openListener)
```

参数	类型	是否必须	说明
context	Context	是	上下文环境
merchantNo	String	是	商户号
memberNo	String	是	会员号
openListener	OnOpenListener	是	回调结果监听

## 发起支付

发起支付流程，同步会检查当前用户的钱包账户开户状态，接口如下：

```
PASCPay.getInstance().pay(@NonNull Context context, @Nullable String merchantNo, @NonNull String memberNo, @NonNull String orderNo, @NonNull OnPayListener listener)
```

参数	类型	是否必须	说明
context	Context	是	上下文环境
merchantNo	String	否	商户号
memberNo	String	是	会员号
mchOrderNo	String	是	商户订单号
listener	OnPayListener	是	支付回调结果监听

####

####统计事件

PASCPay.getInstance().setOnEventListener(OnEventListener listener)

```
public abstract class OnEventListener {
    // label 和 map 有可能是空的
    public abstract void onEvent(@NonNull String eventID, @Nullable String label, @Nullable Map<String, String> map);
    public void onPageStart(String pageName) {
    }
    public void onPageEnd(String pageName) {
    }
    public void onResume(Context context) {
        //Activity
    }
    public void onPause(Context context) {
        //Activity
    }
}
```

## 回调微信支付结果

将微信回调的支付结果通知支付Sdk,由上面的WXPayEntryActivity 调用的, 目前只针对微信。接口如下:

```
PASCPay.getInstance().notifyPayResult (int payType, int payResult, String msg)
```

参数	类型	是否必须	说明
payType	int	是	StatusTable.PayType.WECHAT, StatusTable.PayType.ALIPAY
payResult	int	是	PayResult.PAY_SUCCESS、PayResult.PAY_CANCEL、PayResult.PAY_FAILED、PayResult.PAY_WAITING
msg	String	否	可以为空

## 设置位置信息

在App获取到定位后，调用该Api将位置信息传给支付Sdk，不是必须的，接口如下：

```
PASCPay.getInstance().setLBS(PayLocation payLocation)
```

参数	类型	是否必须	说明
payLocation	PayLocation	是	包含两个字段，longitude, latitude

## 释放资源

该接口目前不调用

```
PASCPay.getInstance().detach(Context context)
```

参数	类型	是否必须	说明
context	Context	是	上下文

## 返回码解析

```
接口
OnPayListener{
    public abstract void onPayResult(int code, String msg)
}
```

code 和 msg 信息如下

```
/**
 * 回调接入方 支付结果
 */
@NotProguard
public final class PASCPayResult {

    /** Result Code Start */
```



```
/**
 * 等待支付结果
 */
public static final int PASC_PAY_CODE_WAITING = 1;

/**
 * 打开钱包首页，或支付成功
 */
public static final int PASC_PAY_CODE_SUCCESS = 0;

/**
 * 初始化失败
 */
public static final int PASC_PAY_CODE_SDK_INIT_FAILED = -1;

/**
 * sdk鉴权未通过
 */
public static final int PASC_PAY_CODE_SDK_AUTH_FAILED = -2;

/**
 * 用户取消
 */
public static final int PASC_PAY_CODE_CANCELED = -3;

/**
 * 支付失败
 */
public static final int PASC_PAY_CODE_FAILED = -4;

/**
 * Token失效
 */
public static final int PASC_PAY_CODE_TOKEN_INVALID = -5;

/**
 * 参数为空
 */
public static final int PASC_PAY_CODE_PARAM_ERROR = -6;

/** Result Code End */

/** Result Msg Start */

/**
 * 支付成功
 */
```

```

public static final String PASC_PAY_MSG_SUCCESS = "支付成功";

/**
 * 等待支付结果
 */
public static final String PASC_PAY_MSG_WAITING = "等待支付结果";

/**
 * 初始化失败
 */
public static final String PASC_PAY_MSG_SDK_INIT_FAILED = "Sdk初始化失败";

/**
 * Sdk鉴权未通过
 */
public static final String PASC_PAY_MSG_SDK_AUTH_FAILED = "Sdk鉴权未通过";

/**
 * 用户取消
 */
public static final String PASC_PAY_MSG_CANCELED = "用户取消";

/**
 * 支付失败
 */
public static final String PASC_PAY_MSG_FAILED = "支付失败";

/**
 * Token失效
 */
public static final String PASC_PAY_MSG_TOKEN_INVALID = "Token失效";

/**
 * 参数有误
 */
public static final String PASC_PAY_MSG_PARAM_ERROR = "请检查参数";

/** Result Msg End */

}

```

## 代码混淆配置

务必完整添加如下混淆配置：

## 支付Sdk代码混淆

```
-keep public class * extends android.view.View {  
    public (android.content.Context);  
    public (android.content.Context, android.util.AttributeSet);  
    public (android.content.Context, android.util.AttributeSet, int);  
}  
  
-keep class com.tencent.mm.opensdk.* {;}  
-keep class com.tendcloud.tenddata.* {;}
```