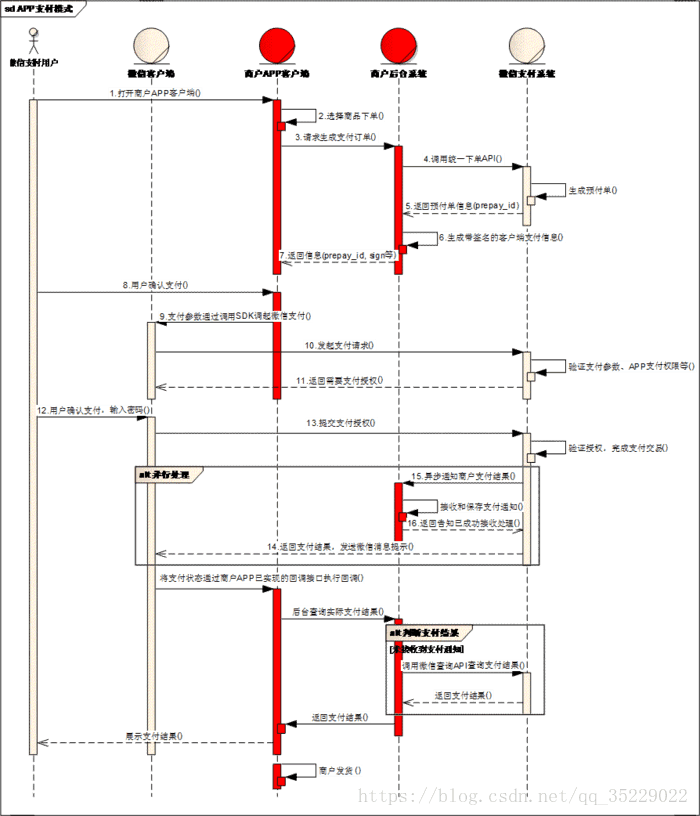
前言

在App中接入微信支付其实很简单，大部分工作主要都是后台来完成，客户端只需要调用后台，得到后台返回的字符串后在调用微信的api即可。但是会有些坑，稍不注意就会掉坑里，下面会讲到。

一、接入流程

在微信开发平台上有详细介绍，这个就不多说了。

交互时序图



商户系统和微信支付系统主要交互说明:

1.用户在商户App中选择商品，提交订单，选择微信支付。

2商户后台收到用户支付单，调用微信支付统一下单接口(后台处理，客户端不需要做)。

3统一下单接口返回正常的prepay\_id,再按签名规范重新生成签名后，将数据传输给App。参与签名的字段为appid,partnerid,paepayid,nonceStr,timeStamp,package.注意:package的值格式为Sign=WXPay。客户端拿到这些字段后就可以发起支付了。

二.APP端开发步骤

1.导入sdk包。微信支付SDK包

2.在调用api之前需要向微信注册你的appid，代码如下:

{ IWXAPI api; api= WXAPIFactory.createWXAPI(this, null); IWXAPI msgApi = WXAPIFactory.createWXAPI(context, null);

//xxx为你的appid

msgApi.registerApp("xxx");}

1

2

3

3.调用微信api

调后台接口，将后台返回的字段做如下处理:

{ PayReq req = new PayReq();

req.packageValue = "package"; req.sign = "sign"; req.timeStamp = "timestamp"; req.appId = "appid"; req.nonceStr = "noncestr"; req.partnerId = "partnerid"; req.prepayId = "prepayid"; api.sendReq(req); }

1

2

3

这些字段最好是在后台获取，不要放在本地。

4.在androidManifest.xml中配置以下权限:

<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"/>

1

2

3

4

5

开发者可参考微信支付官方Demo

5.支付结果回调

参照微信支付demo，在net.sourceforge.simcpux.wxapi包路径中实现

WXPayEntryActivity类(包名或类名不一致会造成无法回调)，在WXPayEntryActivity类中实现onResp函数，支付完成后，微信App会返回到商户App中并回调onResp函数，开发者需要在该函数中接收通知，判断返回错误码，如果支付成功则去后台查询支付结果再展示用户实际支付结果。注意一定不能以客户端返回作为用户支付的结果，应以服务器端的接收的支付通知或查询API返回的结果为准。代码示例如下:

publicvoidonResp(BaseRespresp){ if(resp.getType()==ConstantsAPI.COMMAND\_PAY\_BY\_WX){ Log.d(TAG,"onPayFinish,errCode="+resp.errCode); AlertDialog.Builderbuilder=newAlertDialog.Builder(this); builder.setTitle(R.string.app\_tip); }

1

2

然后再配置文件中注册WXPayEntryActivity

<activity android:name=".wxapi.WXPayEntryActivity" android:label="@string/app\_name" android:exported="true" android:launchMode= "singleTop" />

1

2

3

返回值:

0:成功

-1：失败，可能原因:签名错误、未注册APPID 项目设置APPID不正确、注册的APPID与设置的不匹配、其他异常等。

–2:用户取消，无需处理。发生场景:用户不支付了，点击取消，返回App。

三.开发中遇到的问题

1 回调结果返回-1，首先检查appid是否正确，然后就是签名问题了，app的签名必须是正式打包好的签名，而且必须和微信开放平台填的一样。



应用签名的keystory要固定，否则签名会不一样。

安装打包好的apk就可以调用微信了但是在编译器run的时候却又还是返回-1，这是因为编译器run的是debug版的签名，和正式的不一样，当然会错了。但是每次调试都要打包一次吗?这岂不是很麻烦，其实只要在build.grade配置几行代码即可搞定:

signingConfigs{ release{storeFile file(“C:/Users/dell/Desktop/xxx.jks”) //该路径为keystore的保存路径 //以下3个根据你的keystore自行配置 storePassword “xxx” keyAlias “xxx” keyPassword “xxx”}

buildTypes { release { minifyEnabled false proguardFiles getDefaultProguardFile(‘proguard-android.txt’), ‘proguard-rules.pro’} debug{ signingConfig signingConfigs.release}

这样一来就可直接run了，非常方便调试。

2回调类名和包名

回调类名必须是WXPayEnteryActivity，而且在该类中要实现WXAPIEventHandler接口

包名必须是你app包名+wxapi。例如com.test.main.wxapi

WXPayEnterActivity必须放到该路径下。

如果与上述条件不一致很可能造成支付后无法回调。如果你不想回调，那么上述包名和类名可不创建，微信支付后会默认返回当前支付的页面。如果想要支付成功后跳转到别的页面，那么则要创建。