

微信支付开发文档

1.微信支付功能

微信支付平台是微信运营商提供给商家用于收支的支付平台,商家可以通过微信运营商提供的微信支付的 API,进行应用程序对接。微信支付目前可以对接超市扫码机、商家微信公众号、商家 PC 平台、商家 APP、商家 H5 程序、以及在微信平台开发的微信小程序。微信支付公众号目前有关于微信支付的相关介绍,微信支付公众号二维码如图 1 所示。



图-1:微信公众号二维码

2.微信支付申请流程

1) 注册帐号

注册公众平台(mp.weixin.qq.com),选择帐号类型为服务号,填写相关资料并通过微信支付认证。

2) 填写资料

商户需提供以下 3 项资料:

1. 经营类目以及对应经营资质
2. 企业联系信息
3. 企业银行账户等信息

其他信息诸如企业法人信息、营业执照、组织机构代码证等将直接从微信公众号认证资料中获取，无需重新填写。

3) 商户验证

在资料提交后，微信支付会向您的结算账户中打一笔数额随机的验证款。待资料审核通过后，查收款项，登录商户平台(pay.weixin.qq.com)，填写款项数额。数额正确即可通过验证。

4) 签署协议

验证通过后，在线签署线上协议

3.微信支付类型

微信支付目前提供的支付方式如图 2 所示。

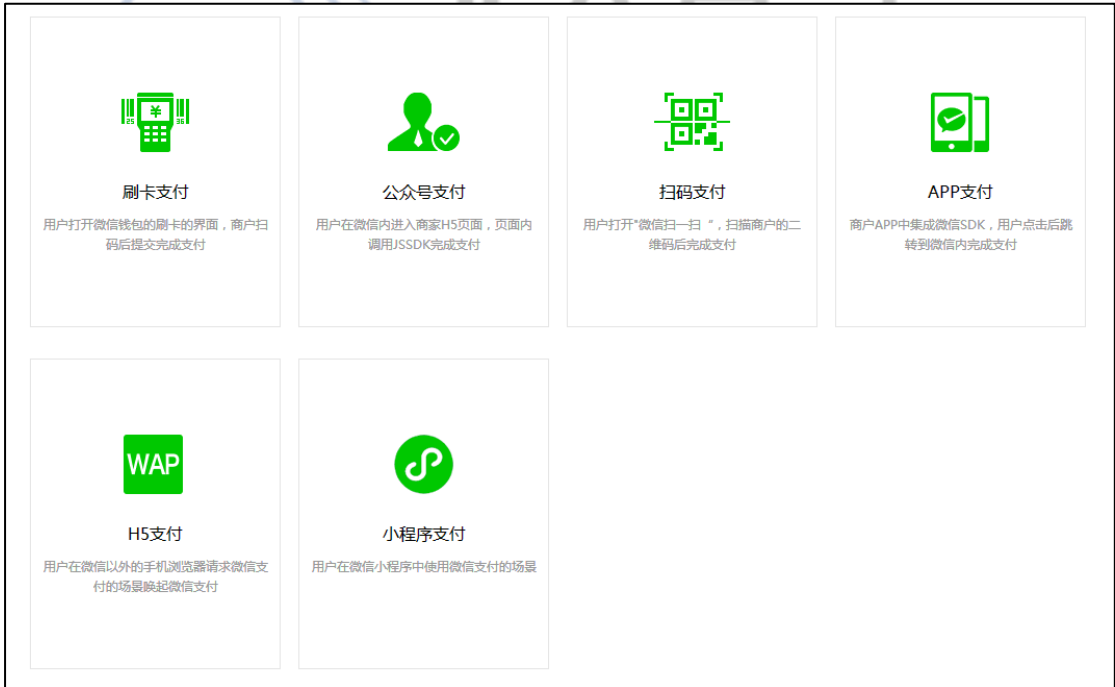


图-2:微信支付方式

本次爱旅行项目，由于是 Web 项目，因此本次系统中选择对接微信扫码支付。

5.微信扫码支付

1) 微信扫码支付效果

微信扫码支付，效果如图-3 所示。



图-3:微信扫码支付效果

2) 微信扫码支付流程

微信扫码支付包括两个模式分别为“模式 1”和“模式 2”,本系统中采用“模式 2”支付机制。关于“模式 1”支付机制，可以通过微信支付官网进行相应了解。微信扫码支付模式 1 的支付时序图如图-4 所示。

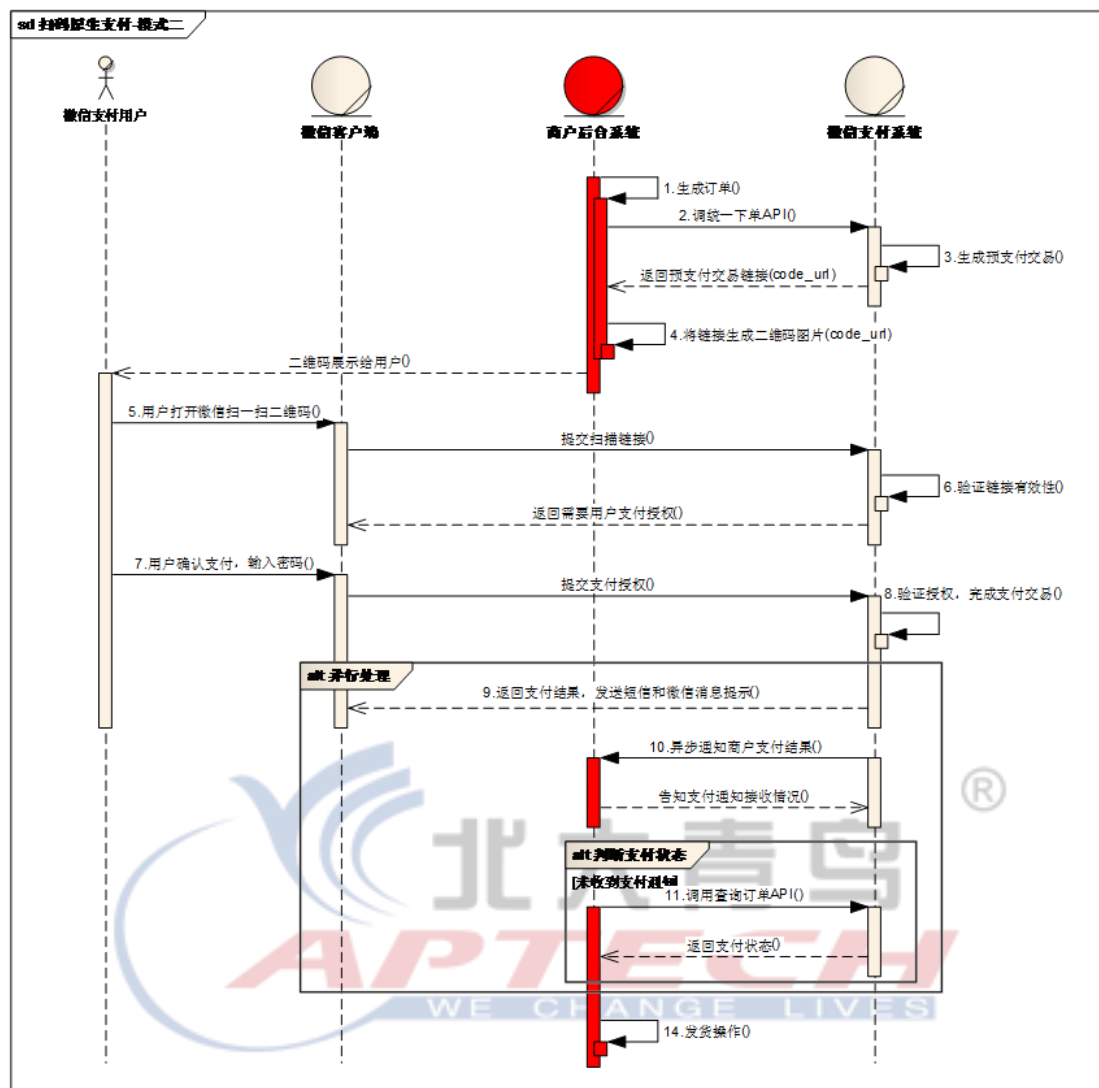


图-4:微信扫码支付模式 1 时序图

微信扫码支付流程解读:

- 1) 商户后台系统根据微信支付规定格式生成二维码（规则见下文），展示给用户扫码。
- 2) 用户打开微信“扫一扫”扫描二维码，微信客户端将扫码内容发送到微信支付系统。
- 3) 微信支付系统收到客户端请求，发起对商户后台系统支付回调 URL 的调用。调用请求将带 productid 和用户的 openid 等参数，并要求商户系统返回数据包,详细请见"本节 3.1 回调数据输入参数"
- 4) 商户后台系统收到微信支付系统的回调请求，根据 productid 生成商户系统的订单。
- 5) 商户系统调用微信支付【统一下单 API】请求下单，获取交易会话标识（prepay_id）
- 6) 微信支付系统根据商户系统的请求生成预支付交易，并返回交易会话标识(prepay_id)。
- 7) 商户后台系统得到交易会话标识 prepay_id（2 小时内有效）。
- 8) 商户后台系统将 prepay_id 返回给微信支付系统。返回数据见"本节 3.2 回调数据输出参

数"

- 9) 微信支付系统根据交易会话标识，发起用户端授权支付流程。
- 10) 用户在微信客户端输入密码，确认支付后，微信客户端提交支付授权。
- 11) 微信支付系统验证后扣款，完成支付交易。
- 12) 微信支付系统完成支付交易后给微信客户端返回交易结果，并将交易结果通过短信、微信消息提示用户。微信客户端展示支付交易结果页面。
- 13) 微信支付系统通过发送异步消息通知商户后台系统支付结果。商户后台系统需回复接收情况，通知微信后台系统不再发送该单的支付通知。
- 14) 未收到支付通知的情况，商户后台系统调用【查询订单 API】。
- 15) 商户确认订单已支付后给用户发货。

6.相关参数获取

商户在微信公众平台(申请扫码支付、公众号支付)或开放平台(申请 APP 支付)按照相应提示，申请相应微信支付模式。微信支付工作人员审核资料无误后开通相应的微信支付权限。微信支付申请审核通过后，商户在申请资料填写的邮箱中收取到由微信支付小助手发送的邮件，此邮件包含开发时需要使用的支付账户信息，见图-5 所示。



图-5:微信支付邮件

注：如果在程序开发中商户账号，还没有申请通过，相关软件工程师可以通过微信平台下载的微信支付的 demo 中使用的账号进行测试。亲测，目前可用！.微信支付 demo 下载地址：https://pay.weixin.qq.com/wiki/doc/api/native.php?chapter=11_1。

7.微信支付安全规范

A. 微信签名算法

第一步：字符串排序

设所有发送或者接收到的数据为集合 M，将集合 M 内非空参数值的参数按照参数名 ASCII 码从小到大排序（字典序），使用 URL 键值对的格式（即 key1=value1&key2=value2...）拼接成字符串 stringA。

第二步：字符串加密

在 stringA 最后拼接上 key 得到 stringSignTemp 字符串，并对 stringSignTemp 根据加密方式进行加密运算，再将得到的字符串所有字符转换为大写，得到 sign 值 signValue。

key 设置路径：微信商户平台(pay.weixin.qq.com)-->账户设置-->API 安全-->密钥设置

注意：

- 参数名 ASCII 码从小到大排序（字典序）；
- 如果参数的值为空不参与签名；
- 参数名区分大小写；
- 验证调用返回或微信主动通知签名时，传送的 sign 参数不参与签名，将生成的签名与该 sign 值作校验。
- 微信接口可能增加字段，验证签名时必须支持增加的扩展字段

B. 生成随机数算法

微信支付 API 接口协议中包含字段 nonce_str，主要保证签名不可预测。我们推荐生成随机数算法如下：调用随机数函数生成，将得到的值转换为字符串。

C. 商户证书

微信扫码支付，暂不需要商户证书进行认证，了解微信商户证书可通过微信支付平台。

8.微信扫码支付开发步骤

1.下载微信支付 Ddemo

微信支付的 demo 源码如图-6 所示，其中包括微信支付提供的相关接口和微信支付的相关实现，以及测试用例。

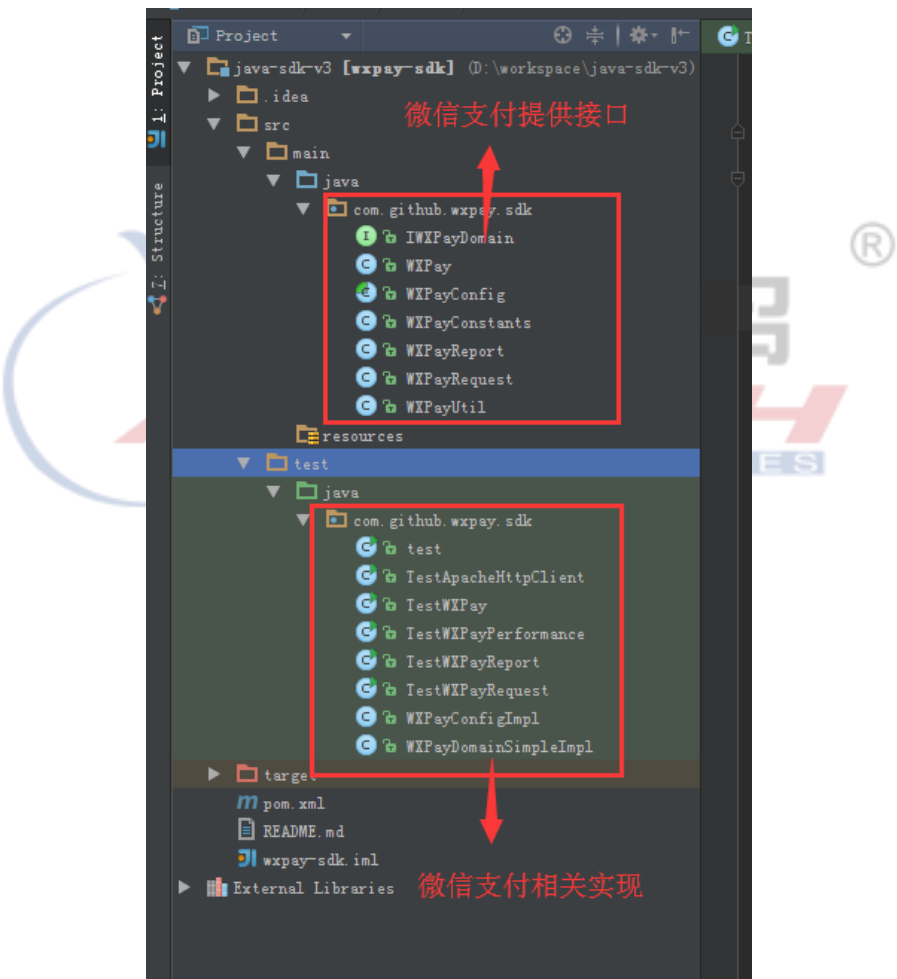


图-6: 微信支付 Demo 结构

注：运行 `TestWXPAY.java`，调用 `doUnifiedOrder` 方法，可以在控制台输出相应请求结果。

2.提取微信支付工具类

由于微信平台 and 应用程序的交互过程中，传递的是 XML 数据，因此在 Java 程序的处理

过程中，需要将 XML 转化成为 Java 相关对象（比如 Map），在 demo 中提供的代码中，提供了 WXPAYUtil 工具类，对 XML 的转化功能进行了封装，在开发独自应用程序的过程中，可以直接 copy 进行使用。

3.编写微信支付调用接口

编写微信支付调用接口可以参照 TestWXPAY 的 doUnifiedOrder 方法进行代码设计。

4.生成微信支付二维码

请查阅微信扫码支付前端设计>二维码生成。

5.编写微信异步通知接口

编写接口接收来自于微信的异步调用，当用户支付成功后，微信回调改地址提醒用户进行后续业务操作。

6.编写定时程序检测微信支付结果

请查阅微信扫码支付前端设计>支付结果轮询。

9.微信扫码支付前端设计

1.logo 设计

在待支付页面选择微信支付模块（需增加对应的 Tabs），如图 7，添加完成后在点击切换 Tabs 时，进行数据的判断，修改 state 对象中的 payType 值等于你定义的值。之后点击支付跳转到二维码扫描界面。



图 7

2. 二维码生成

跳转到二维码扫描页面之后我们需要根据订单的编号进行数据的查找,调用后端的接口“`/trade/api/wxpay/createqccode`”获取到微信支付的地址:
`weixin://wxpay/bizpayurl?pr=w4mU4VI`。根据此地址调用 `qrcode(JS` 插件,使用 `npm install qrcode -save` 添加到项目中。)中的 `toDataURL` 方法,生成图片的地址,通过改变 `state` 中的参数地址,实现页面上的二维码显示,生成效果如图 8;



图 8

3. 支付结果轮询

通过手机扫描完成之后,支付,支付成功之后,后台不能重定向页面的地址和指向,

需要我们增加监听函数，监听代码如下：

```
getOrderStatus=()=>{  
  // 清除之前的监听函数  
  clearTimeout(this.timerId)  
  // 发送请求获取请求数据信息  
  getRequest("/trade/api/wxpay/queryorderstatus/" + this.state.orderNo).then(  
    res=>{  
      //判断支付状态  
      if(res.data.orderStatus===2){  
        // 重定向到支付成功页面  
        window.location.hash  
        = '#orderpaystate/?orderNo='+this.state.orderNo+"&id="+res.data.id;  
      }else{  
        // 支付不成功则继续监测  
        this.timerId=setTimeout(this.getOrderStatus,1000)  
      }  
    }  
  )  
}
```

增加上监听代码之后，我们需要使用 `componentDidMount()`调用监听函数，代码如下：

```
componentDidMount() {  
  this.getOrderStatus();  
}
```

到此我们的微信支付界面就完成了。注：实现周期调用监听函数时，不要使用 `setInterval` 函数，要使用 `setTimeout`，因为我们在使用 `setInterval` 设定的周期比较短时，我们第一次请求数据还没有返回就会发送第二次请求，这样会增加 `server` 端的压力，也会使页面也会出现多次刷新的 `bug`;