

支付功能

微信支付测试商户号开发必备

微信小程序接入微信支付（四）：接收支付结果通知与沙箱测试_微信小程序沙箱支付-CSDN博客

微信小程序 实现支付流程_微信小程序支付-CSDN博客

下面这个应该是最有用的

Nodejs实现微信小程序支付功能前后端实践 全新的微信支付 APIv3_node实现微信小程序支付-CSDN博客

微信小程序 完整支付功能实现（企业项目）

[https://www.bilibili.com/video/BV1dh41127Cc?
p=9&spm_id_from=pageDriver&vd_source=09197c2c9b398bbbac68a972c5c96038](https://www.bilibili.com/video/BV1dh41127Cc?p=9&spm_id_from=pageDriver&vd_source=09197c2c9b398bbbac68a972c5c96038)

web 集成沙箱

支付宝沙箱支付 (yuque.com)

[https://www.bilibili.com/video/BV1hc411X7sD/?
spm_id_from=333.999.0.0&vd_source=09197c2c9b398bbbac68a972c5c96038](https://www.bilibili.com/video/BV1hc411X7sD/?spm_id_from=333.999.0.0&vd_source=09197c2c9b398bbbac68a972c5c96038)

uni-app 集成沙箱

[https://www.bilibili.com/video/BV1Q24y137MW/?
p=5&spm_id_from=pageDriver&vd_source=09197c2c9b398bbbac68a972c5c96038](https://www.bilibili.com/video/BV1Q24y137MW/?p=5&spm_id_from=pageDriver&vd_source=09197c2c9b398bbbac68a972c5c96038)

微信支付测试商户号开发必备

▼

Plain Text

```
1 application.yml配置
2 #微信支付信息配置
3 weixin:
4   appid: wx8397f8696b538317    #微信公众账号或开放平台APP的唯一标识
5   partner: 1473426802          #财付通平台的商户账号
6   partnerkey: T6m9iK73b0kn9g5v426MKfHqH7X8rKwb  #财付通平台的商户密钥
7   notifyurl: http://www.itcast.cn  #回调地址
```

10.5.2 微信支付的流程

1. 创建订单

- 请求创建订单的 API 接口：把（订单金额、收货地址、订单中包含的商品信息）发送到服务器
- 服务器响应的结果：订单编号

2. 订单预支付

- 请求订单预支付的 API 接口：把（订单编号）发送到服务器
- 服务器响应的结果：订单预支付的参数对象，里面包含了订单支付相关的必要参数

3. 发起微信支付

- 调用 `uni.requestPayment()` 这个 API，发起微信支付；把步骤 2 得到的“订单预支付对象”作为参数传递给 `uni.requestPayment()` 方法
- 监听 `uni.requestPayment()` 这个 API 的 `success`，`fail`，`complete` 回调函数

```
level:
  root: info

# 微信登录相关
weixin:
  login:
    info:
      # 微信小程序APPID
      appid: wx2f823cdc8dfba815
      # 微信小程序API私钥
      appsecret: 0139a6d9e93fb88d3d68e0a8a1b06bd6
      # 微信小程序登录成功回调地址
      redirectUrl: https://www.kuangstudy.com/login/api/wx/callback

  pay:
    info:
      # 微信支付网关
      gateway: https://api.weixin.qq.com/sns/jscode2session
      # 微信支付API 秘钥
      appsecret: 0139a6d9e93fb88d3d68e0a8a1b06bd6
      # 微信商户id
      mchid: 1550648501
      # 微信小程序APPID
      appid: wx2f823cdc8dfba815
      # 支付成功回调地址，如果是微信小程序可以不配置
      notifyPath: https://api.kuangstudy.com/api/pay/callback
```

```

1  计算签名值：对API证书（商户私钥）对待签名串（上面构造的签名串）进行SHA256 with RSA签名，并对签名结果进行Base64编码得到签名值
2
3  /**
4   * 微信支付v3 下单签名值生成
5   * @param {string} pem pem证书名称
6   * @param {string} method 请求方法
7   * @param {string} url 微信小程序下单官方api
8   * @param {number} timestamp 时间戳 秒级
9   * @param {string} nonce_str 随机字符串
10  * @param {Object} order 主体（订单）信息
11  */
12  function createOrderSign(pem,method, url, timestamp, nonce_str, order) {
13      // 签名串
14      let signStr = `${method}\n${url}\n${timestamp}\n${nonce_str}\n${JSON.stringify(
15          order
16      )}\n`;
17      // 读取API证书文件内容 apiclient_key.pem的内容
18      let cert = fs.readFileSync(`./pems/files/${pem}`, "utf-8");
19      // 创建使用 RSA 算法和 SHA-256 散列算法的签名对象
20      let sign = crypto.createSign("RSA-SHA256");
21      // 对签名串进行加密处理
22      sign.update(signStr);
23      return sign.sign(cert, "base64");
24  }
25
26  设置HTTP头
27  微信支付商户API v3要求请求通过HTTP Authorization头来传递签名。Authorization由认证类型和签名信息两个部分组成。
28
29  Authorization: 认证类型 签名信息
30
31  具体组成为：
32
33  认证类型，目前为 WECHATPAY2-SHA256-RSA2048
34
35  签名信息
36
37  发起请求的商户（包括直连商户、服务商或渠道商）的商户号mchid
38  商户API证书序列号serial_no，用于声明所使用的证书（apiclient_key.pem 里面的序列号（获取方法有很多 https://www.yesdotnet.com/archive/post/1621531570.html））
39  请求随机串nonce_str
40  时间戳timestamp
41  签名值signature
42  // 生成随机字符串
43  function generateNonceStr(len) {
44      let data = "ABCDEFGHJKMNPQRSTWXYZabcdefhijkmnprstxyz2345678";
45      let str = "";
46      for (let i = 0; i < len; i++) {
47          str += data.charAt(Math.floor(Math.random() * data.length));
48      }
49      return str;
50  }

```

```

51
52 let timestamp = Math.floor(new Date().getTime() / 1000);
53 let nonce_str = generateNonceStr(32);
54 // 计算签名值
55 let signature = createOrderSign(
56     API证书名称,
57     "POST",
58     "/v3/pay/transactions/jsapi",
59     timestamp,
60     nonce_str,
61     wxOrderInfo
62 );
63 // 设置HTTP头
64 let Authorization = `WECHATPAY2-SHA256-RSA2048 mchid="${ac.mchid}",nonce_str
    ="${nonce_str}",timestamp="${timestamp}",signature="${signature}",serial_no
    ="${商户API证书序列号}"`;

65
66 正式请求 预支付接口
67 /**
68  * 微信支付v3 支付信息获取交易会会话标识 prepay_id
69  * @param {Object} order 主体信息
70  * @param notifyUrl 回调地址 https://qy.xxx.com/v1/payment/wx/success 下面有具体实
    现方式
71  */
72 exports.getPrepayInfo = async function (order, notifyUrl) {
73     let timestamp = Math.floor(new Date().getTime() / 1000);
74     let nonce_str = generateNonceStr(32);
75     const ac = await getThirdKeys()
76     let wxOrderInfo = {
77         mchid: 商户号,
78         appid: 小程序appid,
79         notify_url: notifyUrl, // 回调地址 这里需要我们自行实现用来接收支付结果信息
80         out_trade_no: order.orderNo, // 上面创建的订单的订单号 我们自己自定义的
81         description: order.description, // 商品描述
82         amount: {
83             total: order.amount, // 单位为分
84             currency: "CNY"
85         },
86         payer: {
87             openid: order.openid // 用户的openid
88         }
89     }
90     let signature = createOrderSign(
91         ac.pem,
92         "POST",
93         "/v3/pay/transactions/jsapi",
94         timestamp,
95         nonce_str,
96         wxOrderInfo
97     );
98     let Authorization = `WECHATPAY2-SHA256-RSA2048 mchid="${ac.mchid}",nonce_str
    ="${nonce_str}",timestamp="${timestamp}",signature="${signature}",serial_no="${a
    c.serial_no}"`;

99
100     // 拿到 "prepay_id": "wx26112221580621e9b071c00d9e993b00666"
101

```

```
102     return await axios.post("https://api.mch.weixin.qq.com/v3/pay/transactions/j
103     sapi", wxOrderInfo, {
104         headers: {Authorization: Authorization},
105     })
106 }
107
108
109 后端生成支付参数
110 后端生成支付参数响应给前端小程序进行拉起支付
111
112 /**
113  * 微信支付v3 付款签名生成支付参数
114  * @param {string} prepay_id 预支付交易会话标识
115  */
116 exports.createPaySign = async function (prepay_id) {
117     let timeStamp = (Math.floor(new Date().getTime() / 1000)).toString();
118     let nonceStr = generateNonceStr(32);
119     const ac = await getThirdKeys()
120     let signStr = `${ac.appid}\n${timeStamp}\n${nonceStr}\nprepay_id=${prepay_i
121     d}\n`;
122     let cert = fs.readFileSync(`./pems/files/${ac.pem}`, "utf-8");
123     let sign = crypto.createSign("RSA-SHA256");
124     sign.update(signStr);
125     return {
126         paySign: sign.sign(cert, "base64"),
127         timestamp: timeStamp,
128         nonce_str: nonceStr,
129         signType: 'RSA',
130         package: 'prepay_id=' + prepay_id
131     };
132 }
```