

# 一、微信支付介绍

## 1、微信扫码支付申请

微信扫码支付是商户系统按微信支付协议生成支付二维码，用户再用微信“扫一扫”完成支付的模式。该模式适用于PC网站支付、实体店单品或订单支付、媒体广告支付等场景。

申请步骤：（了解）

### 第一步：注册公众号（类型须为：**服务号**）

请根据营业执照类型选择以下主体注册：[个体工商户](#) | [企业/公司](#) | [政府](#) | [媒体](#) | [其他类型](#)。

### 第二步：认证公众号

公众号认证后才可申请微信支付，认证费：300元/年。

### 第三步：提交资料申请微信支付

登录公众平台，点击左侧菜单【微信支付】，开始填写资料等待审核，审核时间为1-5个工作日内。

### 第四步：开户成功，登录商户平台进行验证

资料审核通过后，请登录联系人邮箱查收商户号和密码，并登录商户平台填写财付通备付金打的小额资金数额，完成账户验证。

### 第五步：在线签署协议

本协议为线上电子协议，签署后方可进行交易及资金结算，签署完立即生效。

## 2、开发文档

### 微信支付接口调用的整体思路：

按API要求组装参数，以XML方式发送（POST）给微信支付接口（URL），微信支付接口也是以XML方式给予响应。程序根据返回的结果（其中包括支付URL）生成二维码或判断订单状态。

在线微信支付开发文档：

<https://pay.weixin.qq.com/wiki/doc/api/index.html>

- (1) appid: 微信公众账号或开放平台APP的唯一标识
- (2) mch\_id: 商户号（配置文件中的partner）
- (3) partnerkey: 商户密钥

(4) sign:数字签名, 根据微信官方提供的密钥和一套算法生成的一个加密信息, 就是为了保证交易的安全性

## 3、微信支付SDK

SDK与DEMO下载

平台和语言	说明	支付模式	操作
JAVA	【微信支付】API对应的SDK和调用示例	付款码支付、JSAPI支付、Native支付	<a href="#">下载</a>
.NET C#	【微信支付】API对应的SDK和调用示例	付款码支付、JSAPI支付、Native支付	<a href="#">下载</a>
PHP	【微信支付】API对应的SDK和调用示例	付款码支付、JSAPI支付、Native支付	<a href="#">下载</a>

我们主要会用到微信支付SDK的以下功能:

### (1) 获取随机字符串

```
WXPAYUtil.generateNonceStr()
```

### (2) MAP转换为XML字符串 (自动添加签名)

```
WXPAYUtil.generateSignedXml(param, partnerkey)
```

### (3) XML字符串转换为MAP

```
WXPAYUtil.xmlToMap(result)
```

## 二、微信支付开发

### 1、微信支付接口-生成微信支付二维码

场景: 用户扫描商户展示在各种场景的二维码进行支付

使用案例:

线下: 家乐福超市、7-11便利店、上品折扣线下店等

线上: 大众点评网站、携程网站、唯品会、美丽说网站等

## 开发模式：

**模式一：** 商户在后台给你生成二维码，用户打开扫一扫

**模式二：** 商户后台系统调用微信支付统一下单API生成预付交易，将接口返回的链接生成二维码，用户扫码后输入密码完成支付交易。注意：该模式的预付单有效期为**2**小时，过期后无法支付。

**微信支付：生成xml发送请求**

**操作模块： service\_orders**

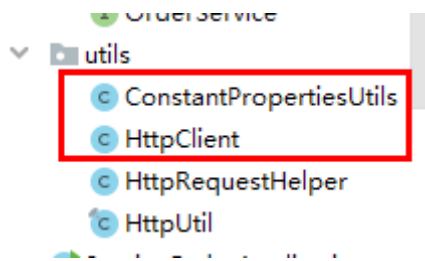
### (1) service\_orders引入依赖

```
1 <dependency>
2     <groupId>com.github.wxpay</groupId>
3     <artifactId>wxpay-sdk</artifactId>
4     <version>0.0.3</version>
5 </dependency>
```

### (2) 添加配置

```
1 spring.redis.host=192.168.44.165
2 spring.redis.port=6379
3 spring.redis.database= 0
4 spring.redis.timeout=1800000
5 spring.redis.lettuce.pool.max-active=20
6 spring.redis.lettuce.pool.max-wait=-1
7 #最大阻塞等待时间(负数表示没限制)
8 spring.redis.lettuce.pool.max-idle=5
9 spring.redis.lettuce.pool.min-idle=0
10
11 #关联的公众号appid
12 weixin.pay.appid=wx74862e0dfcf69954
13 #商户号
14 weixin.pay.partner=1558950191
15 #商户key
16 weixin.pay.partnerkey=T6m9iK73b0kn9g5v426MKfHQH7X8rKwb
```

### (3) 复制工具类



### (4) 添加添加交易记录方法

#### 添加PaymentMapper接口

```
1 public interface PaymentMapper extends BaseMapper<PaymentInfo> {  
2 }
```

#### 添加PaymentService接口

```
1 public interface PaymentService extends IService<PaymentInfo> {  
2     /**  
3      * 保存交易记录  
4      * @param order  
5      * @param paymentType 支付类型（1：微信 2：支付宝）  
6      */  
7     void savePaymentInfo(OrderInfo order, Integer paymentType);  
8 }
```

#### 添加PaymentServiceImpl实现类

```
1 @Service  
2 public class PaymentServiceImpl extends  
3     ServiceImpl<PaymentMapper, PaymentInfo> implements PaymentService {  
4     /**  
5      * 保存交易记录  
6      * @param orderInfo  
7      * @param paymentType 支付类型（1：微信 2：支付宝）  
8      */  
9     @Override
```

```

10     public void savePaymentInfo(OrderInfo orderInfo, Integer paymentType) {
11         QueryWrapper<PaymentInfo> queryWrapper = new QueryWrapper<>();
12         queryWrapper.eq("order_id", orderInfo.getId());
13         queryWrapper.eq("payment_type", paymentType);
14         Integer count = baseMapper.selectCount(queryWrapper);
15         if(count >0) return;
16         // 保存交易记录
17         PaymentInfo paymentInfo = new PaymentInfo();
18         paymentInfo.setCreateTime(new Date());
19         paymentInfo.setOrderId(orderInfo.getId());
20         paymentInfo.setPaymentType(paymentType);
21         paymentInfo.setOutTradeNo(orderInfo.getOutTradeNo());
22         paymentInfo.setPaymentStatus(PaymentStatusEnum.UNPAID.getStatus());
23         String subject = new DateTime(orderInfo.getReserveDate()).toString("
24         paymentInfo.setSubject(subject);
25         paymentInfo.setTotalAmount(orderInfo.getAmount());
26         baseMapper.insert(paymentInfo);
27     }
28 }

```

## (5) 添加支付service接口与实现

### 添加WeixinService接口

```

1 public interface WeixinService {
2     /**
3      * 根据订单号下单，生成支付链接
4      */
5     Map createNative(Long orderId);
6 }

```

### 添加WeixinServiceImpl实现类

```

1 @Service
2 public class WeixinServiceImpl implements WeixinService {
3
4     @Autowired
5     private OrderService orderService;
6     @Autowired

```

```

7     private PaymentService paymentService;
8     @Autowired
9     private RedisTemplate redisTemplate;
10    /**
11     * 根据订单号下单，生成支付链接
12     */
13    @Override
14    public Map createNative(Long orderId) {
15        try {
16            Map payMap = (Map) redisTemplate.opsForValue().get(orderId.toString());
17            if(null != payMap) return payMap;
18            //根据id获取订单信息
19            OrderInfo order = orderService.getById(orderId);
20            // 保存交易记录
21            paymentService.savePaymentInfo(order, PaymentTypeEnum.WEIXIN.getDesc());
22            //1、设置参数
23            Map paramMap = new HashMap();
24            paramMap.put("appid", ConstantPropertiesUtils.APPID);
25            paramMap.put("mch_id", ConstantPropertiesUtils.PARTNER);
26            paramMap.put("nonce_str", WXPUtil.generateNonceStr());
27            Date reserveDate = order.getReserveDate();
28            String reserveDateString = new DateTime(reserveDate).toString("yyyy-MM-dd HH:mm:ss");
29            String body = reserveDateString + "就诊" + order.getDepname();
30            paramMap.put("body", body);
31            paramMap.put("out_trade_no", order.getOutTradeNo());
32            //paramMap.put("total_fee", order.getAmount().multiply(new BigDecimal(100)).intValue());
33            paramMap.put("total_fee", "1");//为了测试
34            paramMap.put("spbill_create_ip", "127.0.0.1");
35            paramMap.put("notify_url", "http://guli.shop/api/order/weixinPay/notifyUrl");
36            paramMap.put("trade_type", "NATIVE");
37            //2、HttpClient来根据URL访问第三方接口并且传递参数
38            HttpClient client = new HttpClient("https://api.mch.weixin.qq.com/pay/unifiedorder");
39            //client设置参数
40            client.setXmlParam(WXPUtil.generateSignedXml(paramMap, ConstantPropertiesUtils.APPID));
41            client.setHttps(true);
42            client.post();
43            //3、返回第三方的数据
44            String xml = client.getContent();
45            Map<String, String> resultMap = WXPUtil.xmlToMap(xml);
46            //4、封装返回结果集
47            Map map = new HashMap<>();
48            map.put("orderId", orderId);
49            map.put("totalFee", order.getAmount());

```

```

50         map.put("resultCode", resultMap.get("result_code"));
51         map.put("codeUrl", resultMap.get("code_url"));
52         if(null != resultMap.get("result_code")) {
53             //微信支付二维码2小时过期，可采取2小时未支付取消订单
54             redisTemplate.opsForValue().set(orderId.toString(), map, 1000);
55         }
56         return map;
57     } catch (Exception e) {
58         e.printStackTrace();
59         return new HashMap<>();
60     }
61 }
62 }

```

## (6) 添加Controller方法

### 添加WeixinController类

```

1  @RestController
2  @RequestMapping("/api/order/weixin")
3  public class WeixinController {
4      @Autowired
5      private WeixinService weixinPayService;
6      /**
7       * 下单 生成二维码
8       */
9      @GetMapping("/createNative/{orderId}")
10     public R createNative(
11         @ApiParam(name = "orderId", value = "订单id", required = true)
12         @PathVariable("orderId") Long orderId) {
13         Map map = weixinPayService.createNative(orderId);
14         return R.ok().data(map);
15     }
16 }

```

## 2、微信支付前端整合

### (1) 在wx.js定义方法

```

1 createNative(orderId) {
2   return request({
3     url: `/api/order/weixin/createNative/${orderId}`,
4     method: 'get'
5   })
6 }

```

## (2) 安装npm install vue-qriously

### (3) 修改order/show.vue组件

```

1 <template>
2   <!-- header -->
3   <div class="nav-container page-component">
4     <!--左侧导航 #start -->
5     <div class="nav left-nav">
6       <div class="nav-item ">
7         <span class="v-link clickable dark" onclick="javascript:window.locat
8       </div>
9       <div class="nav-item selected">
10        <span class="v-link selected dark" onclick="javascript:window.locati
11      </div>
12      <div class="nav-item ">
13        <span class="v-link clickable dark" onclick="javascript:window.locat
14      </div>
15      <div class="nav-item ">
16        <span class="v-link clickable dark"> 修改账号信息 </span>
17      </div>
18      <div class="nav-item ">
19        <span class="v-link clickable dark"> 意见反馈 </span>
20      </div>
21    </div>
22    <!-- 左侧导航 #end -->
23    <!-- 右侧内容 #start -->
24    <div class="page-container">
25      <div class="order-detail">
26        <div class="title"> 挂号详情</div>
27        <div class="status-bar">
28          <div class="left-wrapper">
29            <div class="status-wrapper BOOKING_SUCCESS">
30              <span class="iconfont">□</span> {{ orderInfo.param.orderStatus

```



```
31         </div>
32     </div>
33     <div class="right-wrapper">
34         
36             <div> 微信<span class="iconfont">□</span>关注“预约挂号”</div>
37             <div class="watch-wrapper"> 快速挂号，轻松就医</div>
38         </div>
39     </div>
40 </div>
41 <div class="info-wrapper">
42     <div class="title-wrapper">
43         <div class="block"></div>
44         <div>挂号信息</div>
45     </div>
46     <div class="info-form">
47         <el-form ref="form" :model="form">
48             <el-form-item label="就诊人信息： ">
49                 <div class="content"><span>{{ orderInfo.patientName }}</span>
50             </el-form-item>
51             <el-form-item label="就诊日期： ">
52                 <div class="content"><span>{{ orderInfo.reserveDate }} {{ or
53             </el-form-item>
54             <el-form-item label="就诊医院： ">
55                 <div class="content"><span>{{ orderInfo.hosname }} </span></
56             </el-form-item>
57             <el-form-item label="就诊科室： ">
58                 <div class="content"><span>{{ orderInfo.depname }} </span></
59             </el-form-item>
60             <el-form-item label="医生职称： ">
61                 <div class="content"><span>{{ orderInfo.title }} </span></di
62             </el-form-item>
63             <el-form-item label="医事服务费： ">
64                 <div class="content">
65                     <div class="fee">{{ orderInfo.amount }}元
66                 </div>
67             </div>
68             </el-form-item>
69             <el-form-item label="挂号单号： ">
70                 <div class="content"><span>{{ orderInfo.outTradeNo }} </span>
71             </el-form-item>
72             <el-form-item label="挂号时间： ">
73                 <div class="content"><span>{{ orderInfo.createTime }}</span>
```

```

74         </el-form-item>
75     </el-form>
76 </div>
77 </div>
78 <div class="rule-wrapper mt40">
79     <div class="rule-title"> 注意事项</div>
80     <div>1、请确认就诊人信息是否准确，若填写错误将无法取号就诊，损失由本人承
81         <span style="color:red">2、【取号】就诊当天需在{{ orderInfo.fetchT
82         3、【退号】在{{ orderInfo.quitTime }}前可在线退号 ，逾期将不可办理退
83         4、预约挂号支持自费患者使用身份证预约，同时支持北京市医保患者使用北京社
84         5、请注意北京市医保患者在住院期间不能使用社保卡在门诊取号。
85     </div>
86 </div>
87 <div class="bottom-wrapper mt60" v-if="orderInfo.orderStatus == 0 ||
88     <div class="button-wrapper">
89         <div class="v-button white" @click="cancelOrder()">取消预约</div>
90     </div>
91     <div class="button-wrapper ml20" v-if="orderInfo.orderStatus == 0"
92         <div class="v-button" @click="pay()">支付</div>
93     </div>
94 </div>
95 </div>
96 </div><!-- 右侧内容 #end -->
97 <!-- 微信支付弹出框 -->
98 <el-dialog :visible.sync="dialogPayVisible" style="text-align: left" :ap
99     <div class="container">
100         <div class="operate-view" style="height: 350px;">
101             <div class="wrapper wechat">
102                 <div>
103                     <qriously :value="payObj.codeUrl" :size="220"/>
104                     <div style="text-align: center;line-height: 25px;margin-bottom
105                         请使用微信扫一扫<br/>
106                         扫描二维码支付
107                 </div>
108             </div>
109         </div>
110     </div>
111 </div>
112 </el-dialog>
113 </div><!-- footer -->
114 </template>
115 <script>
116 import '~/assets/css/hospital_personal.css'

```

```
117 import '~/assets/css/hospital.css'
118 import orderInfoApi from '@api/orderinfo'
119 import weixinApi from '@api/wx'
120 export default {
121   data() {
122     return {
123       orderId: null,
124       orderInfo: {
125         param: {}
126       },
127       dialogPayVisible: false,
128       payObj: {},
129       timer: null // 定时器名称
130     }
131   },
132   created() {
133     this.orderId = this.$route.query.orderId
134     this.init()
135   },
136   methods: {
137     init() {
138       orderInfoApi.getOrders(this.orderId).then(response => {
139         this.orderInfo = response.data.orderInfo
140       })
141     },
142     pay() {
143       this.dialogPayVisible = true
144       weixinApi.createNative(this.orderId).then(response => {
145         this.payObj = response.data
146         if(this.payObj.codeUrl == '') {
147           this.dialogPayVisible = false
148           this.$message.error("支付错误")
149         }
150       })
151     }
152   }
153 }
154 </script>
155 <style>
156   .info-wrapper {
157     padding-left: 0;
158     padding-top: 0;
159   }
```

```
160 .content-wrapper {
161     color: #333;
162     font-size: 14px;
163     padding-bottom: 0;
164 }
165 .bottom-wrapper {
166     width: 100%;
167 }
168 .button-wrapper {
169     margin: 0;
170 }
171 .el-form-item {
172     margin-bottom: 5px;
173 }
174 .bottom-wrapper .button-wrapper {
175     margin-top: 0;
176 }
177 </style>
```