

# 一、准备工作

## 1、全局配置的跳转路径

```
1 # 微信开放平台 重定向url
2 wx.open.redirect_url=http://回调地址/api/ucenter/wx/callback
```

## 2、修改当前项目启动端口号为8160

## 3、测试回调是否可用

在WxApiController中添加方法

```
1 @GetMapping("callback")
2 public String callback(String code, String state, HttpSession session) {
3
4     //得到授权临时票据code
5     System.out.println("code = " + code);
6     System.out.println("state = " + state);
7 }
```

# 二、后台开发

## 1、添加依赖

```
1 <!--httpClient-->
2 <dependency>
3     <groupId>org.apache.httpcomponents</groupId>
4     <artifactId>httpClient</artifactId>
5 </dependency>
6 <!--gson-->
```

```
7 <dependency>
8   <groupId>com.google.code.gson</groupId>
9   <artifactId>gson</artifactId>
10 </dependency>
```

## 2、创建httpClient工具类

放入util包

```
1 HttpClientUtils.java
```

## 3、创建回调controller方法

在WxApiController.java中添加如下方法

```
1 @GetMapping("callback")
2 public String callback(String code, String state, HttpSession session) {
3     //第一步 获取临时票据 code
4     System.out.println("code:"+code);
5     //第二步 拿着code和微信id和密钥, 请求微信固定地址 , 得到两个值
6     //使用code和appid以及appsecret换取access_token
7     // %s 占位符
8     StringBuffer baseAccessTokenUrl = new StringBuffer()
9         .append("https://api.weixin.qq.com/sns/oauth2/access_token")
10        .append("?appid=%s")
11        .append("&secret=%s")
12        .append("&code=%s")
13        .append("&grant_type=authorization_code");
14    String accessTokenUrl = String.format(baseAccessTokenUrl.toString(),
15                                          ConstantPropertiesUtil.WX_OPEN_APP_
16                                          ConstantPropertiesUtil.WX_OPEN_APP_
17                                          code);
18    //使用httpClient请求这个地址
19    try {
20        String accesstokenInfo = HttpClientUtils.get(accessTokenUrl);
21        System.out.println("accesstokenInfo:"+accesstokenInfo);
22        //从返回字符串获取两个值 openid 和 access_token
23        JSONObject jsonObject = JSONObject.parseObject(accesstokenInfo);
24        String access_token = jsonObject.getString("access_token");
```

```
25 String openid = jsonObject.getString("openid");
26
27 //判断数据库是否存在微信的扫描人信息
28 //根据openid判断
29 UserInfo userInfo = userInfoService.selectWxInfoOpenId(openid);
30 if(userInfo == null) { //数据库不存在微信信息
31     //第三步 拿着openid 和 access_token请求微信地址，得到扫描人信息
32     String baseUserInfoUrl = "https://api.weixin.qq.com/sns/userinfo"
33         "?access_token=%s" +
34         "&openid=%s";
35     String userInfoUrl = String.format(baseUserInfoUrl, access_token,
36         openid);
37     String resultInfo = HttpClientUtils.get(userInfoUrl);
38     System.out.println("resultInfo:" + resultInfo);
39     JSONObject resultUserInfoJson = JSONObject.parseObject(resultInfo);
40     //解析用户信息
41     //用户昵称
42     String nickname = resultUserInfoJson.getString("nickname");
43     //用户头像
44     String headimgurl = resultUserInfoJson.getString("headimgurl");
45
46     //获取扫描人信息添加数据库
47     userInfo = new UserInfo();
48     userInfo.setNickName(nickname);
49     userInfo.setOpenid(openid);
50     userInfo.setStatus(1);
51     userInfoService.save(userInfo);
52 }
53 //返回name和token字符串
54 Map<String,String> map = new HashMap<>();
55 String name = userInfo.getName();
56 if(StringUtils.isEmpty(name)) {
57     name = userInfo.getNickName();
58 }
59 if(StringUtils.isEmpty(name)) {
60     name = userInfo.getPhone();
61 }
62 map.put("name", name);
63
64 //判断userInfo是否有手机号，如果手机号为空，返回openid
65 //如果手机号不为空，返回openid值是空字符串
66 //前端判断：如果openid不为空，绑定手机号，如果openid为空，不需要绑定手机号
67 if(StringUtils.isEmpty(userInfo.getPhone())) {
68     map.put("openid", userInfo.getOpenid());
69 }
```

```

68         } else {
69             map.put("openid", "");
70         }
71         //使用jwt生成token字符串
72         String token = JwtHelper.createToken(userInfo.getId(), name);
73         map.put("token", token);
74         //跳转到前端页面
75         return "redirect:http://localhost:3000/weixin/callback?token="+map.get("token");
76     } catch (Exception e) {
77         e.printStackTrace();
78         return null;
79     }
80 }

```

## 4、创建UserInfoService方法

根据openid查询用户信息

```

1  //定义方法
2  UserInfo selectWxInfoOpenId(String openid);
3
4  //实现方法
5  @Override
6  public UserInfo selectWxInfoOpenId(String openid) {
7      return baseMapper.selectOne(new QueryWrapper<UserInfo>().eq("openid", openid));
8  }

```

## 5、创建UserInfoServiceImpl的登录方法

绑定手机号操作

```

1  //手机验证码登录
2  @Override
3  public Map<String, Object> login(LoginVo loginVo) {
4      //获取手机验证码登录两个值
5      String phone = loginVo.getPhone();
6      String code = loginVo.getCode();
7      String openid = loginVo.getOpenid();
8

```

```
9 //非空判断
10 if(StringUtils.isEmpty(phone) || StringUtils.isEmpty(code)) {
11     throw new YyghException(20001,"数据为空");
12 }
13
14 //手机验证码校验
15 //从redis获取手机号对应验证码
16 String redisCode = redisTemplate.opsForValue().get(phone);
17 //比较输入验证码和redis验证码是否一样
18 if(!code.equals(redisCode)) {
19     throw new YyghException(20001,"验证码输入不正确");
20 }
21
22 //返回页面显示名称
23 Map<String, Object> map = new HashMap<>();
24 if(StringUtils.isEmpty(openid)) {
25     //根据手机查询数据库
26     QueryWrapper<UserInfo> wrapper = new QueryWrapper<>();
27     wrapper.eq("phone",phone);
28     UserInfo userInfo = baseMapper.selectOne(wrapper);
29     //如果返回对象为空，就是第一次登录，存到数据库登录数据
30     if(userInfo == null) {
31         userInfo = new UserInfo();
32         userInfo.setName("");
33         userInfo.setPhone(phone);
34         userInfo.setStatus(1);
35         baseMapper.insert(userInfo);
36     }
37     //判断用户是否可用
38     if(userInfo.getStatus() == 0) {
39         throw new YyghException(20001,"用户已经禁用");
40     }
41     map = get(userInfo);
42 } else {
43     baseMapper.delete(new QueryWrapper<UserInfo>().eq("phone", phone));
44
45     UserInfo userInfoWx = baseMapper.selectOne(new QueryWrapper<UserInfo>());
46     //判断用户是否可用
47     userInfoWx.setPhone(phone);
48     baseMapper.updateById(userInfoWx);
49
50     if(userInfoWx.getStatus() == 0) {
51         throw new YyghException(20001,"用户已经禁用");
```

```

52     }
53     map = get(userInfo.wx);
54 }
55 return map;
56 }
57
58 private Map<String, Object> get(UserInfo userInfo) {
59     //返回页面显示名称
60     Map<String, Object> map = new HashMap<>();
61     String name = userInfo.getName();
62     if(StringUtils.isEmpty(name)) {
63         name = userInfo.getNickName();
64     }
65     if(StringUtils.isEmpty(name)) {
66         name = userInfo.getPhone();
67     }
68     map.put("name", name);
69     //根据userid和name生成token字符串
70     String token = JwtHelper.createToken(userInfo.getId(), name);
71     map.put("token", token);
72     return map;
73 }

```

## 6、回调返回页面

说明：我们只期望返回一个空页面，然后跟登录层通信就可以了，其实就是一个过渡页面，所以我们要给这个过渡页面定义一个空模板

### (1) 添加空模板组件：/layouts/empty.vue

```

1 <template>
2 <div>
3 <nuxt/>
4 </div>
5 </template>

```

### (2) 创建回调返回页面 /pages/weixin/callback.vue

```

1 <template>
2 <!-- header -->
3 <div>

```

```

4   </div>
5   <!-- footer -->
6 </template>
7 <script>
8 export default {
9   layout: "empty",
10  data() {
11    return {
12    }
13  },
14  mounted() {
15    let token = this.$route.query.token
16    let name = this.$route.query.name
17    let openid = this.$route.query.openid
18    // 调用父vue方法
19    window.parent['loginCallback'](name, token, openid)
20  }
21 }
22 </script>

```

### (3) 在myheader.vue添加方法

```

1  mounted() {
2    // 注册全局登录事件对象
3    window.loginEvent = new Vue();
4    // 监听登录事件
5    loginEvent.$on('loginDialogEvent', function () {
6      document.getElementById("loginDialog").click();
7    })
8    // 触发事件，显示登录层: loginEvent.$emit('loginDialogEvent')
9    //初始化微信js
10   const script = document.createElement('script')
11   script.type = 'text/javascript'
12   script.src = 'https://res.wx.qq.com/connect/zh_CN/htmledition/js/wxLogin.'
13   document.body.appendChild(script)
14
15   // 微信登录回调处理
16   let self = this;
17   window["loginCallback"] = (name,token, openid) => {
18     self.loginCallback(name, token, openid);
19   }
20 },

```

```

21
22 loginCallback(name, token, openid) {
23     // 打开手机登录层，绑定手机号，改逻辑与手机登录一致
24     if(openid !== '') {
25         this.userInfo.openid = openid
26         this.showLogin()
27     } else {
28         this.setCookies(name, token)
29     }
30 },

```

## 7、预约挂号页面调整

修改/pages/hospital/\_hoscode.vue组件

### (1) 引入cookie

```

1 import cookie from 'js-cookie'

```

### (2) 修改方法

```

1 schedule(decode) {
2     // 登录判断
3     let token = cookie.get('token')
4     if (!token) {
5         loginEvent.$emit('loginDialogEvent')
6         return
7     }
8     window.location.href = '/hospital/schedule?hoscode=' + this.hospital.hoscode
9 }

```