

一、准备工作

<https://open.weixin.qq.com>

1、注册

2、邮箱激活

3、完善开发者资料

4、开发者资质认证

准备营业执照，1-2个工作日审批、300元

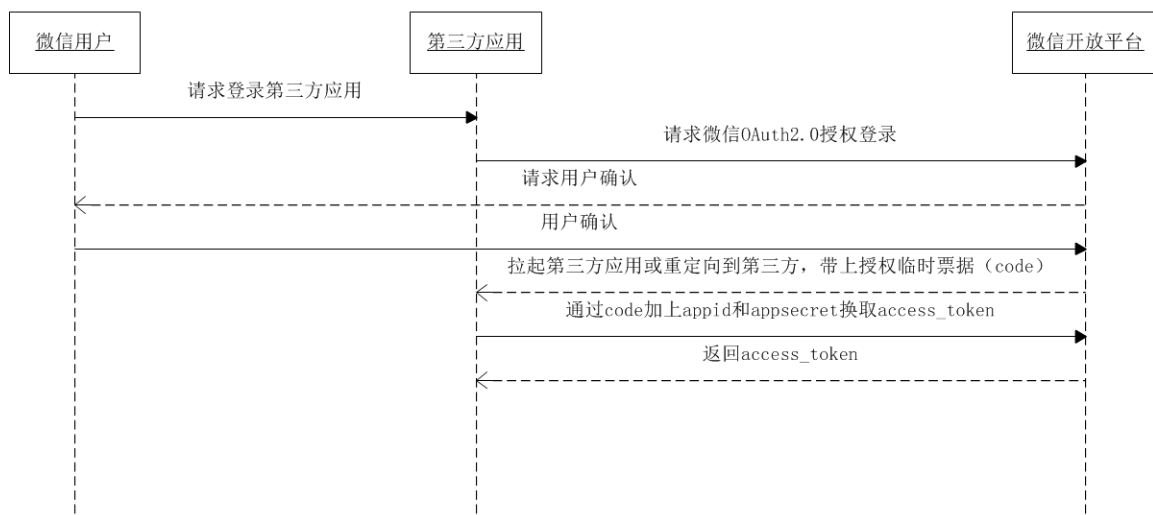
5、创建网站应用

提交审核，7个工作日审批

6、熟悉微信登录流程

参考文档：https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419316505&token=e547653f995d8f402704d5cb2945177dc8aa4e7e&lang=zh_CN

获取access_token时序图



二、后端开发service_user

说明：微信登录二维码我们是以弹出层的形式打开，不是以页面形式，所以做法是不一样的，参考如下链接，上面有相关弹出层的方式

https://developers.weixin.qq.com/doc/oplatform/Website_App/WeChat_Login/Wechat_Login.html

如图：

步骤1：在页面中先引入如下JS文件（支持https）：

```
http://res.wx.qq.com/connect/zh_CN/htmledition/js/wxLogin.js
```

步骤2：在需要使用微信登录的地方实例以下JS对象：

```
var obj = new WxLogin({
  self_redirect:true,
  id:"login_container",
  appid: "",
  scope: "",
  redirect_uri: "",
  state: "",
  style: "",
  href: ""
});
```

因此我们的操作步骤为：

第一步我们通过接口把对应参数返回页面；

第二步在头部页面启动打开微信登录二维码；

第三步处理登录回调接口；

第四步回调返回页面通知微信登录层回调成功

第五步如果是第一次扫描登录，则绑定手机号码，登录成功

接下来我们根据步骤，一步一步实现

1、添加配置

application.properties添加相关配置信息

```
1 # 微信开放平台 appid
2 wx.open.app_id=你的appid
3 # 微信开放平台 appsecret
4 wx.open.app_secret=你的appsecret
5 # 微信开放平台 重定向url
6 wx.open.redirect_url=http://你的服务器名称/api/ucenter/wx/callback
```

引入依赖

```
1 <dependencies>
2     <dependency>
3         <groupId>org.apache.httpcomponents</groupId>
4         <artifactId>httpclient</artifactId>
5     </dependency>
6 </dependencies>
```

2、创建常量类

创建util包，创建ConstantPropertiesUtil.java常量类

```
1 @Component
2 //@PropertySource("classpath:application.properties")
3 public class ConstantPropertiesUtil implements InitializingBean {
4
5     @Value("${wx.open.app_id}")
6     private String appId;
7
8     @Value("${wx.open.app_secret}")
9     private String appSecret;
10
11     @Value("${wx.open.redirect_url}")
12     private String redirectUrl;
13
14     public static String WX_OPEN_APP_ID;
15     public static String WX_OPEN_APP_SECRET;
16     public static String WX_OPEN_REDIRECT_URL;
17
18     @Override
19     public void afterPropertiesSet() throws Exception {
```

```

20         WX_OPEN_APP_ID = appId;
21         WX_OPEN_APP_SECRET = appSecret;
22         WX_OPEN_REDIRECT_URL = redirectUrl;
23     }
24 }

```

3、创建controller

service_user微服务中创建api包

api包中创建WxApiController, 返回微信参数

```

1  @Controller
2  @RequestMapping("/api/ucenter/wx")
3  public class WeixinApiController {
4
5      @Autowired
6      private UserInfoService userInfoService;
7      @Autowired
8      private RedisTemplate redisTemplate;
9
10     /**
11      * 获取微信登录参数
12      */
13     @GetMapping("getLoginParam")
14     @ResponseBody
15     public R genQrConnect(HttpSession session) throws UnsupportedEncodingException {
16         String redirectUri = URLEncoder.encode(ConstantPropertiesUtil.WX_OPEN_REDIRECT_URL, "UTF-8");
17         Map<String, Object> map = new HashMap<>();
18         map.put("appid", ConstantPropertiesUtil.WX_OPEN_APP_ID);
19         map.put("redirectUri", redirectUri);
20         map.put("scope", "snsapi_login");
21         map.put("state", System.currentTimeMillis()+""); //System.currentTimeMillis()
22         return R.ok().data(map);
23     }
24 }

```

授权url参数说明

参数	是否必须	说明
----	------	----

参数	是否必须	说明
appid	是	应用唯一标识
redirect_uri	是	请使用urlencode对链接进行处理
response_type	是	填code
scope	是	应用授权作用域，拥有多个作用域用逗号（,）分隔，网页应用目前仅填写snsapi_login即
state	否	用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止csrf攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加session进行校验

三、前端显示登录二维码

1、封装api请求

创建api/wx.js

```

1 import request from '@/utils/request'
2
3 const api_name = `/api/ucenter/wx`
4
5 export default {
6   getLoginParam() {
7     return request({
8       url: `${api_name}/getLoginParam`,
9       method: `get`
10    })
11  }
12 }
```

2、修改组件

修改layouts/myheader.vue文件，添加微信二维码登录逻辑

```
1 // 1、引入api
2 import weixinApi from '@api/weixin'
3
4 //2、引入微信js
5 mounted() {
6     // 注册全局登录事件对象
7     window.loginEvent = new Vue();
8     // 监听登录事件
9     loginEvent.$on('loginDialogEvent', function () {
10         document.getElementById("loginDialog").click();
11     })
12     // 触发事件，显示登录层: loginEvent.$emit('loginDialogEvent')
13     //初始化微信js
14     const script = document.createElement('script')
15     script.type = 'text/javascript'
16     script.src = 'https://res.wx.qq.com/connect/zh_CN/htmledition/js/wxLogin.'
17     document.body.appendChild(script)
18
19     // 微信登录回调处理
20     let self = this;
21     window["loginCallback"] = (name,token, openid) => {
22         self.loginCallback(name, token, openid);
23     }
24 },
25
26 //3、实例化微信JS对象
27 //添加微信登录方法
28 loginCallback(name, token, openid) {
29     // 打开手机登录层，绑定手机号，改逻辑与手机登录一致
30     if(openid != '') {
31         this.userInfo.openid = openid
32         this.showLogin()
33     } else {
34         this.setCookies(name, token)
35     }
36 },
37
38 weixinLogin() {
39     this.dialogAttr.showLoginType = 'weixin'
40
41     weixinApi.getLoginParam().then(response => {
42         var obj = new WxLogin({
43             self_redirect:true,
```

```
44     id: 'weixinLogin', // 需要显示的容器id
45     appid: response.data.appid, // 公众号appid wx*****
46     scope: response.data.scope, // 网页默认即可
47     redirect_uri: response.data.redirectUri, // 授权成功后回调的url
48     state: response.data.state, // 可设置为简单的随机数加session用来校验
49     style: 'black', // 提供"black"、"white"可选。二维码的样式
50     href: '' // 外部css文件url, 需要https
51   })
52 })
53 },
54 //说明: 微信登录方法已绑定weixinLogin(), 查看页面
```