

微信开放平台（针对开发者和公司）

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

1、准备工作

- 网站应用微信登录是基于OAuth2.0协议标准构建的微信OAuth2.0授权登录系统。
- 在进行微信OAuth2.0授权登录接入之前，在微信开放平台注册开发者帐号，并拥有一个已审核通过的网站应用，并获得相应的AppID和AppSecret，申请微信登录且通过审核后，可开始接入流程。
- 注册帐号和申请应用都是免费的，必须要有一个线上的网站，才能审核通过（过程还是挺麻烦的），就可以使用微信的登录了
- 但是如果使用微信支付的功能，就必须认证开发者资质（认证一次300块人民币）

2、名词解释

2.1 OAuth2.0协议

OAuth(Open Authorization)协议就是为用户资源的授权提供了一个安全、开放、简易的标准。

OAuth在第三方应用与服务提供商之间设置了一个授权层，第三方应用通过授权层获取令牌，再通过令牌获取信息。

比如：皇宫大院，并不是每一块区域你都可以去溜达的。一个小奴才专门负责打扫后宫的寝室卫生，后宫门口有N多带刀侍卫，每次进门工作。都要问皇上，因为想进入到后宫内院，只有皇帝一个人说的算，皇帝让谁进，谁才能进。但是每次问皇上呢，又太累，所以“令牌”出现了，皇上命人制作了一些令牌，给打扫卫生的小太监每人一个，想进去，出示令牌给侍卫即可。这就是“宫廷版OAuth协议”。

- 玩抖音，发视频，抖音需要访问你相册的授权，话筒的授权，地理位置的授权等等
- 一句话，我不想帐号密码给第三方应用，但我还想用他们的功能，而他们的功能需要我的部分数据来协助。ok，咱玩令牌。
- 令牌与密码的作用都可以进入系统，但是有三点差异：
 - 1、**令牌是短期的**，到期会自动失效，用户自己无法修改。密码一般长期有效，用户不修改，就不会发生变化。
 - 2、**令牌可以被数据所有者撤销**，会立即失效。以上例而言，屋主可以随时取消快递员的令牌。密码一般不允许被他人撤销
 - 3、**令牌有权限范围**，比如只能进小区的二号门。对于网络服务来说，只读令牌就比读写令牌更安全。密码一般是完整权限。
- 上面这些设计，保证了令牌既可以让第三方应用获得权限，同时又随时可控，不会危及系统安全。
- OAuth的四种授权模式：
 - 1、授权码模式（功能最完整、流程最严密的授权模式）

- 说白了，授权码模式，不再client和user之间商量授权，而是client想要被授权，所以client去找了一个和事佬大妈，大妈将client和user叫到了一起（认证服务器），给大妈个面子，这事就这么定了。就是这样的一个过程，全程中认证服务器会发布一个认证码贯穿始终。
- 2、密码模式 - 了解
- 3、简化模式 - 了解
- 4、客户端模式 - 了解

2.2 AppID

- 应用ID，唯一标识（身份证号）

2.3 AppSecret

- 应用的密钥（密码）

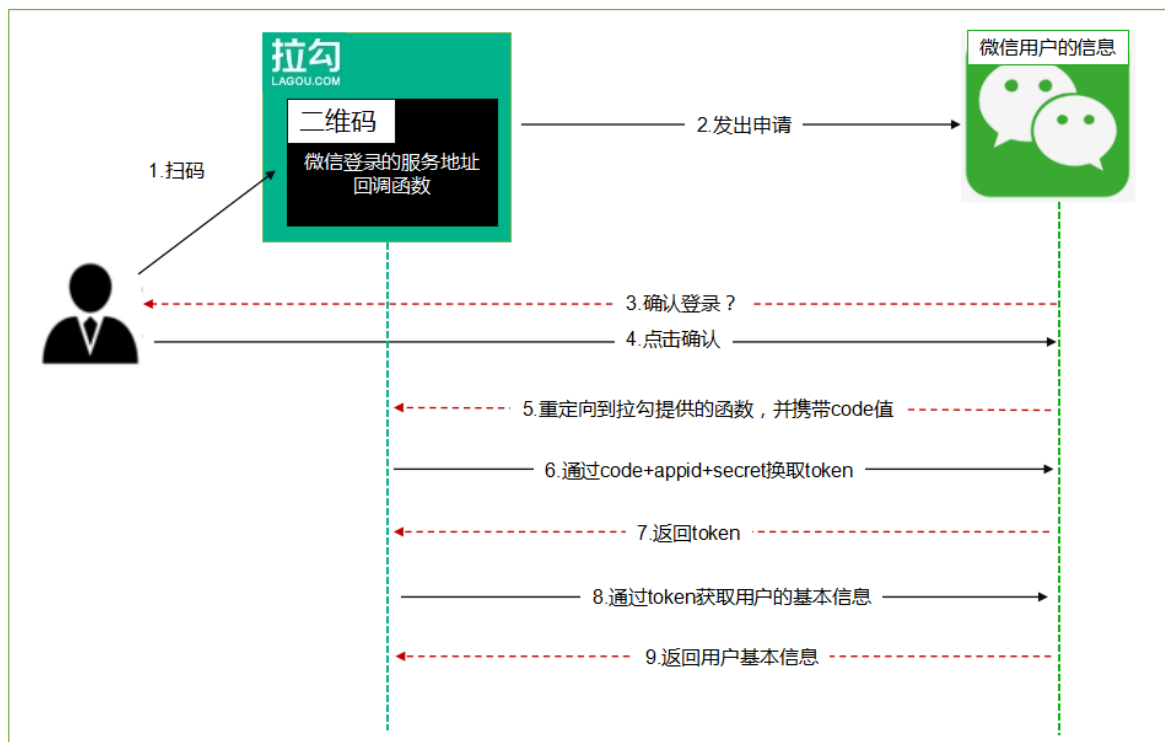
2.4 code

- 授权的临时凭证（例如：临时身份证）

2.5 access_token

- 接口调用凭证（例如：真正的身份证，虎符，令牌）

3、登录授权时序图



4、开发步骤

4.1 vue项目安装

- 微信官方提供的生成二维码的js

```
npm install vue-wxlogin
```

如果不是vue的项目，可以直接引用官方提供的js文件，来生成二维码

4.2 页面引入

```
<div id="loginForm">
  <!-- 登录表单 -->
  <el-form>
    <!-- 登录按钮 -->
    <el-button/>
    <!-- 微信登录图标 -->
    <img/>
  </div>

<!-- 二维码 -->
<wxlogin id="wxLoginForm" style="display:none"
  :appid="appid" :scope="scope" :redirect_uri="redirect_uri">
</wxlogin>

<script>
import wxlogin from 'vue-wxlogin'; // 引入

export default {
  name: "Header",
  components: {
    wxlogin // 声明
  },
  data() {
    return {
      appid:"wxd99431bbff8305a0", // 应用唯一标识，在微信开放平台提交应用审核通过后获得
      scope:"snsapi_login", // 应用授权作用域，网页应用目前仅填写snsapi_login即可
      redirect_uri:"http://www.pinzhi365.com/wxlogin", //重定向地址，（回调地址）
    };
  },
  methods: {
    goToLoginWX() {
      document.getElementById("loginForm").style.display = "none";
      document.getElementById("wxLoginForm").style.display = "block";
    }
  }
}
</script>
```

4.3 修改hosts文件

文件位置：C:\Windows\System32\drivers\etc\hosts

回调默认指定的是80端口，别忘记将tomcat的8003端口修改成80

```
127.0.0.1 www.pinzhi365.com
```

4.4 依赖

```
<!-- 需要使用HttpServletRequest获得参数 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.4</version>
    <scope>provided</scope>
</dependency>
<!-- 需要使用HttpClient发出请求 -->
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.12</version>
</dependency>
```

4.5 封装HttpClient

```
package commons;

import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.utils.URIBuilder;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

import java.net.URI;
import java.util.Map;

/**
 * @BelongsProject: lagou-edu-web
 * @Author: GuoAn.Sun
 * @CreateTime: 2020-09-22 11:30
 * @Description: httpclient的封装工具类
 */
public class HttpClientUtil {

    public static String doGet(String url) {
        return doGet(url, null);
    }

    /**
     * get请求，支持request请求方式，不支持restfull方式
     *
     * @param url 请求地址
     * @param param 参数
     * @return 响应的字符串
     */
}
```

```

    */
    public static String doGet(String url, Map<String, String> param) {
        // 创建httpClient对象
        CloseableHttpClient httpClient = HttpClients.createDefault();

        String resultString = "";
        CloseableHttpResponse response = null;
        try {
            // 创建url
            URIBuilder builder = new URIBuilder(url);
            if (param != null) {
                // 在url后面拼接请求参数
                for (String key : param.keySet()) {
                    builder.addParameter(key, param.get(key));
                }
            }
            URI uri = builder.build();

            // 创建http get请求
            HttpGet httpGet = new HttpGet(uri);
            // 执行请求
            response = httpClient.execute(httpGet);
            // 从响应对象中获取状态码（成功或失败的状态）
            int statusCode = response.getStatusLine().getStatusCode();
            System.out.println("响应的状态 = " + statusCode);
            // 200表示响应成功
            if (statusCode == 200) {
                // 响应的内容字符串
                resultString = EntityUtils.toString(response.getEntity(), "UTF-8");
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // 释放资源
            try {
                if (response != null) {
                    response.close();
                }

                httpClient.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return resultString;
    }
}

```

4.6 定义从微信返回的数据对象

```
public class Token {
    private String access_token;//接口调用凭证
    private String expires_in; //access_token接口调用凭证超时时间，单位（秒）
    private String refresh_token;//用户刷新access_token
    private String openid; //授权用户唯一标识
    private String scope; //用户授权的作用域，使用逗号（,）分隔
    private String unionid; //当且仅当该网站应用已获得该用户的userinfo授权时，才会出现该字段。
```

```
public class User {
    private String openid;//普通用户的标识，对当前开发者帐号唯一
    private String nickname;//普通用户昵称
    private String sex;//普通用户性别，1为男性，2为女性
    private String province;//普通用户个人资料填写的省份
    private String city;//普通用户个人资料填写的城市
    private String country;//国家，如中国为CN
    private String headimgurl;//用户头像，最后一个数值代表正方形头像大小（有0、46、64、96、132数值可选，0代表640*640正方形头像），用户没有头像时该项为空
    private String privilege;//用户特权信息，json数组，如微信沃卡用户为（chinaunicom）
    private String unionid;//用户统一标识。针对一个微信开放平台帐号下的应用，同一用户的unionid是唯一的。
```

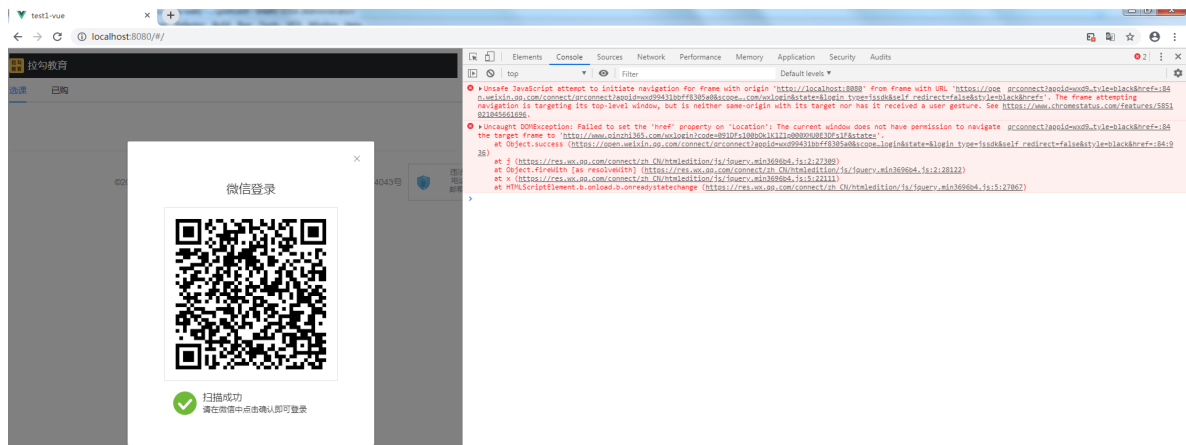
4.7 回调函数controller

```
@RestController
public class TestWX {

    @RequestMapping("wxlogin")
    public Object wxlogin(HttpServletRequest request) throws IOException {
        // 1.通过扫码去微信服务器请求，返回的code值
        String code = request.getParameter("code");
        System.out.println("code = " + code);
        // 2.通过code获取access_token，定义获取access_token的url
        String getTokenUrl = "https://api.weixin.qq.com/sns/oauth2/access_token?appid=wxid99431bbff8305a0&secret=60f78681d063590a469f1b297feff3c4&code="+code+"&grant_type=authorization_code";
        // 3.发请求并获得access_token
        String access_token = HttpClientUtil.doGet(getTokenUrl);
        System.out.println("access_token = "+access_token);
        // 4.将返回的json格式字符串转换成对象
        Token token = JSON.parseObject(access_token, Token.class);
        // 5.通过access_token获取获取微信用户个人信息
        String getUserUrl = "https://api.weixin.qq.com/sns/userinfo?access_token="+token.getAccess_token()+"&openid="+token.getOpenid();
        // 6.发请求并获取用户信息
        String user_token = HttpClientUtil.doGet(getUserUrl);
        // 7.将返回的json格式字符串转换成对象
        User user = JSON.parseObject(user_token, User.class);
        // 8.自定义网页逻辑
        System.out.println("昵称 = " + user.getNickname());

        return user;
    }
}
```

如果下面的错误：是因为谷歌浏览器有bug，待修复！切换别的浏览器即可



web服务的端口号必须是80！

4.8 后续

WxLoginController

```
@RestController
@RequestMapping("user")
public class WxLoginController {

    @Reference // 远程消费
    private UserService userService;

    private UserDTO dto = null; // 是否用微信登录成功，dto为null，则尚未登录

    @GetMapping("wxlogin")
    public String wxlogin(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // 1. 微信官方发给我们一个临时凭证
        String code = request.getParameter("code");
        System.out.println("【临时凭证】code = " + code);
        // 2. 通过code，去微信官方申请一个正式的token（令牌）
        String getTokenByCode_url =
"https://api.weixin.qq.com/sns/oauth2/access_token?
appid=wxid99431bbff8305a0&secret=60f78681d063590a469f1b297feff3c4&code=" + code +
"&grant_type=authorization_code";
        String tokenString = HttpClientUtil.doGet(getTokenByCode_url);
        System.out.println("tokenString = " + tokenString);
        // 将json格式的token字符串转换成实体对象，方便存和取
        Token token = JSON.parseObject(tokenString, Token.class);

        // 3. 通过token，去微信官方获取用户的信息
        String getUserByToken_url = "https://api.weixin.qq.com/sns/userinfo?
access_token=" + token.getAccess_token() + "&openid=" + token.getOpenid();
        String userinfoString = HttpClientUtil.doGet(getUserByToken_url);
        System.out.println("userinfoString = " + userinfoString);
        // 将json格式的user字符串转换成实体对象，方便存和取
        WxUser wxUser = JSON.parseObject(userinfoString, WxUser.class);
        System.out.println("微信昵称 = " + wxUser.getNickname());
        System.out.println("微信头像 = " + wxUser.getHeadimgurl());
    }
}
```

```

        // 拉勾的业务流程！ 需要 手机号（wxUser.getUnionid()）和密码
        (wxUser.getUnionid()),头像和昵称

        User user = null;
        dto = new UserDTO();
        // 检测手机号是否注册
        Integer i = userService.checkPhone(wxUser.getUnionid());
        if(i == 0){
            // 未注册，自动注册并登录
            userService.register(wxUser.getUnionid(),
wxUser.getUnionid(),wxUser.getNickname(),wxUser.getHeadimgurl());
            dto.setMessage("手机号尚未注册，系统已帮您自动注册，请牢记密码！");
            user = userService.login(wxUser.getUnionid(), wxUser.getUnionid());
        }else{
            user = userService.login(wxUser.getUnionid(), wxUser.getUnionid());
            if(user == null){
                dto.setState(300); //300表示失败
                dto.setMessage("帐号密码不匹配，登录失败！");
            }else{
                dto.setState(200); //200表示成功
                dto.setMessage("登录成功！");
            }
        }

        dto.setContent(user);

        response.sendRedirect("http://localhost:8080");
        return null;
    }

    @GetMapping("checkwxStatus")
    public UserDTO checkwxStatus(){
        return this.dto;
    }

    @GetMapping("logout")
    public Object logout(){
        this.dto = null;
        return null;
    }
}

```

Header.vue

```

<script>
created(){

    // 当刷新页面，组件创建成功之后，立刻检测本地储存中是否存在用户对象
    this.userDTO = JSON.parse( localStorage.getItem("user") );
    if(this.userDTO != null){
        this.isLogin = true; // 已登录
    }else{
        // 去检测微信是否登录过
    }
}

```



```

        this.axios
        .get("http://localhost:80/user/checkwxStatus")
        .then( (result)=>{
            this.userDTO = result.data;
            this.phone = this.userDTO.content.phone;
            this.password = this.userDTO.content.password;
            this.login();// 走普通登录
        })
        .catch( (error)=>{
            //this.$message.error("登录失败! ");
        });
    }
}

//登出
logout(){
    localStorage.setItem("user", null); // 将登录成功的对象信息保存到本地储存中
    this.isLogin = false; // 更新登录状态
    alert('谢谢使用，再见! ');
    // 去检测微信是否登录过
    this.axios.get("http://localhost:80/user/logout")
        .then( (result)=>{
        })
        .catch( (error)=>{
            //this.$message.error("登录失败! ");
        });
    }
}
</script>

```

4.9 解决二维码在谷歌浏览器的bug

谷歌浏览器调试的时候，iframe标签跨域问题导致无法跳转的bug

如果iframe未添加sandbox属性，或者sandbox属性不赋值，就代表采用默认的安全策略

即：iframe的页面将会被当做一个独立的源，并且不能提交表单，不能执行javascript脚本，也不能让包含iframe的父页面导航到其他地方，所有的插件，如flash等也全部不能起作用

简单来说iframe就只剩下一个展示数据的功能，正如他的名字一样，所有的内容都被放进了一个“单独的沙盒”

- sandbox包含的属性及作用：

属性值	说明
allow-scripts	允许运行执行脚本
allow-top-navigation	允许iframe能够主导window.top进行页面跳转
allow-same-origin	允许同域请求,比如ajax,storage
allow-forms	允许进行提交表单
allow-popups	允许iframe中弹出新窗口,比如>window.open,target="_blank"
allow-pointer-lock	在iframe中可以锁定鼠标,主要和鼠标锁定有关

- 加上 sandbox="allow-scripts allow-top-navigation allow-same-origin" 属性,即可解决
- 官方js: http://res.wx.qq.com/connect/zh_CN/htmledition/js/wxLogin.js
- 无法修改微信服务器上的js文件,所以我们将js代码放在本地并进行修改:

```

created(){
    !(function(a, b, c) {
        function d(a) {
            var c = "default";
            a.self_redirect === !0
                ? (c = "true")
                : a.self_redirect === !1 && (c = "false");
            var d = b.createElement("iframe"),
                e =
                    "https://open.weixin.qq.com/connect/qrconnect?appid=" +
                    a.appid +
                    "&scope=" +
                    a.scope +
                    "&redirect_uri=" +
                    a.redirect_uri +
                    "&state=" +
                    a.state +
                    "&login_type=jssdk&self_redirect=" +
                    c +
                    "&styletype=" +
                    (a.styletype || "") +
                    "&sizetype=" +
                    (a.sizetype || "") +
                    "&bgcolor=" +
                    (a.bgcolor || "") +
                    "&rst=" +
                    (a.rst || "");
            (e += a.style ? "&style=" + a.style : ""),
                (e += a.href ? "&href=" + a.href : ""),
                (d.src = e),
                (d.frameBorder = "0"),
                (d.allowTransparency = "true"),
                (d.sandbox = "allow-scripts allow-top-navigation allow-same-origin"),
                // 允许多种请求
                (d.scrolling = "no"),
                (d.width = "300px"),
                (d.height = "400px");
            var f = b.getElementById(a.id);
            (f.innerHTML = ""), f.appendChild(d);
        }
    })(this, document, true);
}

```

```

    }
    a.wxLogin = d;
  })(window, document);
}

```

Course.vue

```
<div id="wxLoginForm"></div>
```

```

methods: {
  // 微信登录
  goToLoginWX() {
    // 普通的登录表单隐藏
    document.getElementById("loginForm").style.display = "none";
    // 显示二维码的容器
    document.getElementById("wxLoginForm").style.display = "block";
    // 去生成二维码
    this.$nextTick(function(){
      this.createCode(); // 直接调用会报错: TypeError: Cannot read property
      'appendChild' of null
    });
  },
  // 生成二维码
  createCode(){
    var obj = new WxLogin({
      id:"wxLoginForm", // 显示二维码的容器
      appid: "wxd99431bbff8305a0", // 应用唯一标识, 在微信开放平台提交应用审核通过后
      scope: "snsapi_login", // 应用授权作用域, 网页应用目前仅填写snsapi_login即可
      redirect_uri: "http://www.pinzhi365.com/wxlogin", // 重定向地址, (回调地址)
      href: "data:text/css;base64,加密后的样式"
    });
  },
}

```

```

.impowerBox .qrcode {width: 200px;}
.impowerBox .title {display: none;}
.impowerBox .info {width: 200px;}
.status_icon {display: none}cs
.impowerBox .status {text-align: center;}

```

我们用站长工具对样式代码进行base64加密：<http://tool.chinaz.com/Tools/Base64.aspx>

The screenshot shows the 'Base64' tab selected in the tool's navigation bar. The input text on the left is:

```

.impowerBox .qrcode {width: 200px;}
.impowerBox .title {display: none;}
.impowerBox .info {width: 200px;}
.status_icon {display: none}cs
.impowerBox .status {text-align: center;}

```

The output on the right is a single line of Base64 encoded text. At the bottom, there are buttons for '多行' (Multi-line), 'Base64加密' (Base64 Encode), 'Base64解密' (Base64 Decode), and '清空结果' (Clear Result). A red arrow points to the 'Base64加密' button.