



链滴

公众号菜单网页授权功能

作者: [nianxingyan](#)

原文链接: <https://ld246.com/article/1495191434859>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

公众号菜单通过网页授权实现view中url动态跳转

官方

官方文档参照：在[官方开发者文档](#)中 --> 微信网页授权项（要下班啦，官方文档上的说明不描述啦，这里仅显示实现方法及代码demo）

公众号菜单配置

1. 获取菜单配置接口：https://api.weixin.qq.com/cgi-bin/get_current_selfmenu_info?access_token=ACCESS_TOKEN access_token由自己存储在服务器端

2. 菜单配置方式: 使用Postman工具进行配置

请求: POST

Headers: Content-Type: application/json

Body:(raw型 JSON)

```
{
  "button":[
    { "type":"click",
      "name":"孕妇课堂",
      "key":"click_knowledges"
    },
    { "type":"view",
      "name":"关于我们",
      "url" : " www.baidu.com" } ]
}
```

3. 将我们的view点击后的希望跳转地址进行urlEncode编码，

urlEncode编码地址：<http://tool.chinaz.com/tools/urlencode.aspx>

我们自己写的微信菜单接口地址，如：test.com/project1/servlet1/view

编码后为：zhilehuo.com%2fproject1%2fservlet1%2fview

4. view中url填地址：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=SCOPE&state=STATE#wechat_redirect 若提示“该链接无访问”，请检查参数是否填写错误，是否拥有scope参数对应的授权作用域权限。

5. 编写服务器端处理请求代码：

```
@Controller
@RequestMapping(value = "servlet1/view")
public class MenuServletController extends HttpServlet {
    @RequestMapping
```

```

public void weixinLogin(HttpServletRequest request, HttpServletResponse response) throws E
ception {
    String code = request.getParameter("code"); //获取code
    if (null == code || "".equals(code))
        throw new Exception();
    //这一步就是拼写微信api请求地址并通过微信的appid 和 微信公众号的AppSecret 以及我们获取到
    针对用户授权回调的code 拿到 这个用户的 openid
    String requestUrl = "https://api.weixin.qq.com/sns/oauth2/access_token?appid=APPID&secre
t=APPSECRET&code=CODE&grant_type=authorization_code".replace("APPID", 填写微信APPI
).replace("APPSECRET", 填写微信公众号的AppSecret).replace("CODE", code);
    String requestResult = RequestService.getRequest(requestUrl);//自己写一个doGet方法 发送do
et请求
    JSONObject getCodeResultJson = JSON.parseObject(requestResult);//把请求成功后的结果转
成JSON对象
    if(getCodeResultJson.getString("openid") == null) {
        throw new Exception(); //没有拿到openid
    }
    String openid = getCodeResultJson.getString("openid");//拿到openid
    //我们自己的动态地址一般需要用户的openid以及其他由openid算出的用户参数
    .... //得到openid以后我们可以算出链接所需参数
    //发送我们的链接url, 带上参数
    response.sendRedirect(url+ "?openid="+openid+"&par1="+par1+"&par2="+par2);
} catch (Exception e){
    e.printStackTrace();
}
}

```

自己写的doGet方法附上:

```

public class RequestService {
    public static String getRequest(String urlString) throws IOException{
        HttpURLConnection conn = null;
        BufferedReader responseReader = null;
        StringBuffer sb = null;
        try {
            URL url = new URL(urlString);
            conn = (HttpURLConnection) url.openConnection();
            //设置请求属性
            conn.setRequestProperty("Content-Type", "application/json");
            conn.setRequestProperty("Connection", "Keep-Alive");// 维持长连接
            conn.setRequestProperty("Charset", "UTF-8");
            conn.connect();
            if (conn.getResponseCode() == HttpURLConnection.HTTP_OK){
                sb=new StringBuffer();
                String readLine=new String();
                responseReader=new BufferedReader(new InputStreamReader(conn.getInputStream(),
UTF-8"));
                while((readLine=responseReader.readLine())!=null){
                    sb.append(readLine).append("\n");
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {

```

```
    if (responseReader != null){
        responseReader.close();
    }
    if (conn != null) {
        conn.disconnect();
    }
}
if (sb != null) {
    return sb.toString();
}
else {
    return null;
}
}
}
```

6.简单实现了view点击后直接跳转动态链接功能，相比于设置为click事件，点击click后返回一个此形的文本超链接 效果更好。

7.关于此功能的拓展：

除了获取openid，还可以获取很多用户信息【拉取用户信息(需scope为 snsapi_userinfo)】，通过code换取网页授权access_token（请求方法：获取code后，请求以下链接获取access_token：）【注：此网页授权access_token与基础支持的access_token不同】

获取access token后，发送请求：http: GET 可以获取用户信息，包括openid，用户昵称，性别，家，省份，城市，头像等用户信息。（不可以获取用户的微信号）