

小程序端代码：

```
1 <button class='btn' open-type='getUserInfo'  
  bindgetUserinfo='getUserInfo2' >  
2  查看规则  
3 </button>
```

```
1 getUserInfo: function(e) {  
2   console.log(e)  
3   app.globalData.userInfo = e.detail.userInfo  
4   this.setData({  
5     userInfo: e.detail.userInfo,  
6     hasUserInfo: true  
7   })  
8 },  
9 getUserInfo2: function(e){  
10  wx.login({  
11    success(res) {  
12      if (res.code) {  
13        wx.getUserInfo({  
14          success: function (e) {  
15            var jsondata = {  
16              code: res.code,  
17              enxy: e  
18            }  
19            wx.request({  
20              url: 'http://localhost:8080/user/getUserInfo',  
21              data: jsondata  
22            })  
23          }  
24        })  
25      }  
26    }  
27  })  
28 })  
29 }
```

后端接收解析：

```
1 @RequestMapping(value = "/user/getUserInfo")
```

```

2 public void getUserInfo(@RequestParam(value = "code",required = false)String code,
3 @RequestParam(value = "enxy",required = false)String res){
4     userService.parseUserInfo(code,res);
5 }

```

```

1 @Override
2 public JSONObject parseUserInfo(String code,String res) {
3     String sessionKey = getSessionKey(code);
4     JSONObject data = getUserInfo(res,sessionKey);
5     return null;
6 }
7
8 public static String getSessionKey(String code){
9     String appid="wxac4848bde7178ad6";
10    String secret="de9f8eb52081ff50486e2fafa67eabb2";
11    String baseUrl="https://api.weixin.qq.com/sns/jscode2session";
12    String url=baseUrl+"?
appid="+appid+"&secret="+secret+"&js_code="+code+"&grant_type=authorization
_code";
13    HttpUtil httpUtil = new HttpUtil();
14    try {
15        String result = httpUtil.get(url);
16        JSONObject data = new JSONObject(result);
17
18        return data.get("session_key").toString();
19    } catch (IOException e) {
20        logger.error(e.getMessage());
21        return null;
22    }
23 }
24
25 /**
26  * 获取encryptedData,iv
27  * @param res
28  * @return
29  */
30 public static JSONObject getUserInfo(String res,String sessionKey){
31
32    JSONObject jsonObject = new JSONObject(res);
33    String encryptedData = jsonObject.get("encryptedData").toString();

```

```
34 String iv = jsonObject.getString("iv");
35 JSONObject data = decryptionUserInfo(encryptedData, sessionKey, iv);
36
37 return data;
38 }
39
40 /**
41  * 解密
42  * @param encryptedData
43  * @param sessionKey
44  * @param iv
45  * @return
46  */
47 public static JSONObject decryptionUserInfo(String encryptedData, String
sessionKey, String iv) {
48 // 被加密的数据
49 byte[] dataByte = Base64.decode(encryptedData);
50 // 加密秘钥
51 byte[] keyByte = Base64.decode(sessionKey);
52 // 偏移量
53 byte[] ivByte = Base64.decode(iv);
54 try {
55 // 如果密钥不足16位，那么就补足。这个if 中的内容很重要
56 int base = 16;
57 if (keyByte.length % base != 0) {
58 int groups = keyByte.length / base + (keyByte.length % base != 0 ? 1 :
0);
59 byte[] temp = new byte[groups * base];
60 Arrays.fill(temp, (byte) 0);
61 System.arraycopy(keyByte, 0, temp, 0, keyByte.length);
62 keyByte = temp; }
63 // 初始化
64 Security.addProvider(new BouncyCastleProvider());
65 Cipher cipher = Cipher.getInstance("AES/CBC/PKCS7Padding", "BC");
66 SecretKeySpec spec = new SecretKeySpec(keyByte, "AES");
67 AlgorithmParameters parameters =
AlgorithmParameters.getInstance("AES");
68 parameters.init(new IvParameterSpec(ivByte));
69 cipher.init(Cipher.DECRYPT_MODE, spec, parameters);
70 // 初始化
71 byte[] resultByte = cipher.doFinal(dataByte);
```

```
72  if (null != resultByte && resultByte.length > 0) {
73  String result = new String(resultByte, "UTF-8");
74  JSONObject data = new JSONObject(result);
75  return data;
76  }
77  } catch (Exception e) {
78  e.printStackTrace();
79  }
80  return null;
81  }
```