

一、用户登录前端

1、封装api方法

(1) 创建api/userInfo.js

```
1 import request from '@utils/request'
2
3 const api_name = `/api/user`
4
5 export default {
6   login(userInfo) {
7     return request({
8       url: `${api_name}/login`,
9       method: `post`,
10      data: userInfo
11    })
12  }
13 }
```

(2) 创建api/msm.js

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

2、添加登录组件

登录层是一个公共层，因此我们把它放在头部组件里面

修改layouts/myheader.vue文件

```
1 <template>
2   <div class="header-container">
3     <div class="wrapper">
4       <!-- logo -->
5       <div class="left-wrapper v-link selected">
6         尚医通 预约挂号统一平台</span>
8       </div>
9       <!-- 右侧 -->
10      <div class="right-wrapper">
11        <span class="v-link clickable">帮助中心</span>
12        <span v-if="name == ''" class="v-link clickable" @click="showLogin
13        <el-dropdown v-if="name != ''" @command="loginMenu">
14          <span class="el-dropdown-link">
15            {{ name }}<i class="el-icon-arrow-down el-icon--right"></i>
16          </span>
17          <el-dropdown-menu class="user-name-wrapper" slot="dropdown">
18            <el-dropdown-item command="/user">实名认证</el-dropdown-item>
19            <el-dropdown-item command="/order">挂号订单</el-dropdown-item>
20            <el-dropdown-item command="/patient">就诊人管理</el-dropdown-it
21            <el-dropdown-item command="/logout" divided>退出登录</el-dropdo
22          </el-dropdown-menu>
23        </el-dropdown>
24      </div>
25    </div>
26
27    <!-- 登录弹出层 -->
28    <el-dialog :visible.sync="dialogUserFormVisible" style="text-align: left
29      <div class="container">
30
31      <!-- 手机登录 #start -->
32      <div class="operate-view" v-if="dialogAttr.showLoginType === 'phone'
33        <div class="wrapper" style="width: 100%">
34          <div class="mobile-wrapper" style="position: static;width: 70%">
35            <span class="title">{{ dialogAttr.labelTips }}</span>
36            <el-form>
37              <el-form-item>
38                <el-input v-model="dialogAttr.inputValue" :placeholder="di
```

```

39         <span slot="suffix" class="sendText v-link" v-if="dialog
40         <span slot="suffix" class="sendText v-link highlight cli
41     </el-input>
42 </el-form-item>
43 </el-form>
44     <div class="send-button v-button" @click="btnClick()"> {{ dial
45 </div>
46 <div class="bottom">
47     <div class="wechat-wrapper" @click="weixinLogin()"><span
48         class="iconfont icon">□</span></div>
49     <span class="third-text"> 第三方账号登录 </span></div>
50 </div>
51 </div>
52 <!-- 手机登录 #end -->
53
54 <!-- 微信登录 #start -->
55 <div class="operate-view" v-if="dialogAttr.showLoginType === 'weixi
56     <div class="wrapper wechat" style="height: 400px">
57         <div>
58             <div id="weixinLogin"></div>
59         </div>
60         <div class="bottom wechat" style="margin-top: -80px;">
61             <div class="phone-container">
62                 <div class="phone-wrapper" @click="phoneLogin()"><span
63                     class="iconfont icon">□</span></div>
64                 <span class="third-text"> 手机短信验证码登录 </span></div>
65             </div>
66         </div>
67     </div>
68 <!-- 微信登录 #end -->
69
70 <div class="info-wrapper">
71     <div class="code-wrapper">
72         <div><span class="iconfont icon">□</span>微信
74         </div>
75         <div class="code-text"> “快速预约挂号”</div>
76     </div>
77     <div class="wechat-code-wrapper">
80     <div class="code-text"> 扫一扫下载</div>
81     <div class="code-text"> “预约挂号”APP</div>

```

```

82         </div>
83     </div>
84     <div class="slogan">
85         <div>xxxxxx官方指定平台</div>
86         <div>快速挂号 安全放心</div>
87     </div>
88 </div>
89 </div>
90 </el-dialog>
91 </div>
92 </template>
93 <script>
94 import cookie from 'js-cookie'
95 import Vue from 'vue'
96 import userInfoApi from '@api/userinfo'
97 import smsApi from '@api/msm'
98 import hospitalApi from '@api/hosp'
99
100 const defaultDialogAttr = {
101     showLoginType: 'phone', // 控制手机登录与微信登录切换
102
103     labelTips: '手机号码', // 输入框提示
104
105     inputValue: '', // 输入框绑定对象
106     placeholder: '请输入您的手机号', // 输入框placeholder
107     maxLength: 11, // 输入框长度控制
108
109     loginBtn: '获取验证码', // 登录按钮或获取验证码按钮文本
110
111     sending: true, // 是否可以发送验证码
112     second: -1, // 倒计时 second>0 : 显示倒计时 second=0 : 重新发送 second<0 : 倒计时结束
113     clearSmsTime: null // 倒计时定时任务引用 关闭登录层清除定时任务
114 }
115 export default {
116     data() {
117         return {
118             userInfo: {
119                 phone: '',
120                 code: '',
121                 openid: ''
122             },
123
124             dialogUserFormVisible: false,

```

```
125     // 弹出层相关属性
126     dialogAttr:defaultDialogAttr,
127
128     name: '' // 用户登录显示的名称
129   }
130 },
131
132 created() {
133   this.showInfo()
134 },
135
136 mounted() {
137   // 注册全局登录事件对象
138   window.loginEvent = new Vue();
139   // 监听登录事件
140   loginEvent.$on('loginDialogEvent', function () {
141     document.getElementById("loginDialog").click();
142   })
143   // 触发事件，显示登录层: loginEvent.$emit('loginDialogEvent')
144 },
145
146 methods: {
147   // 绑定登录或获取验证码按钮
148   btnClick() {
149     // 判断是获取验证码还是登录
150     if(this.dialogAttr.loginBtn == '获取验证码') {
151       this.userInfo.phone = this.dialogAttr.inputValue
152
153       // 获取验证码
154       this.getCodeFun()
155     } else {
156       // 登录
157       this.login()
158     }
159   },
160
161   // 绑定登录，点击显示登录层
162   showLogin() {
163     this.dialogUserFormVisible = true
164
165     // 初始化登录层相关参数
166     this.dialogAttr = { ...defaultDialogAttr }
167   },
```

```
168
169 // 登录
170 login() {
171     this.userInfo.code = this.dialogAttr.inputValue
172
173     if(this.dialogAttr.loginBtn == '正在提交...') {
174         this.$message.error('重复提交')
175         return;
176     }
177     if (this.userInfo.code == '') {
178         this.$message.error('验证码必须输入')
179         return;
180     }
181     if (this.userInfo.code.length != 4) {
182         this.$message.error('验证码格式不正确')
183         return;
184     }
185     this.dialogAttr.loginBtn = '正在提交...'
186     userInfoApi.login(this.userInfo).then(response => {
187         console.log(response.data)
188         // 登录成功 设置cookie
189         this.setCookies(response.data.name, response.data.token)
190     }).catch(e => {
191         this.dialogAttr.loginBtn = '马上登录'
192     })
193 },
194
195 setCookies(name, token) {
196     cookie.set('token', token, { domain: 'localhost' })
197     cookie.set('name', name, { domain: 'localhost' })
198     window.location.reload()
199 },
200
201 // 获取验证码
202 getCodeFun() {
203     if (!(/^1[34578]\d{9}$/.test(this.userInfo.phone))) {
204         this.$message.error('手机号码不正确')
205         return;
206     }
207
208     // 初始验证码相关属性
209     this.dialogAttr.inputValue = ''
210     this.dialogAttr.placeholder = '请输入验证码'
```

```
211     this.dialogAttr.maxLength = 6
212     this.dialogAttr.loginBtn = '马上登录'
213
214     // 控制重复发送
215     if (!this.dialogAttr.sending) return;
216
217     // 发送短信验证码
218     this.timeDown();
219     this.dialogAttr.sending = false;
220     smsApi.sendCode(this.userInfo.phone).then(response => {
221         this.timeDown();
222     }).catch(e => {
223         this.$message.error('发送失败，重新发送')
224         // 发送失败，回到重新获取验证码界面
225         this.showLogin()
226     })
227 },
228
229 // 倒计时
230 timeDown() {
231     if(this.clearSmsTime) {
232         clearInterval(this.clearSmsTime);
233     }
234     this.dialogAttr.second = 60;
235
236     this.dialogAttr.labelTips = '验证码已发送至' + this.userInfo.phone
237     this.clearSmsTime = setInterval(() => {
238         --this.dialogAttr.second;
239         if (this.dialogAttr.second < 1) {
240             clearInterval(this.clearSmsTime);
241             this.dialogAttr.sending = true;
242             this.dialogAttr.second = 0;
243         }
244     }, 1000);
245 },
246
247 // 关闭登录层
248 closeDialog() {
249     if(this.clearSmsTime) {
250         clearInterval(this.clearSmsTime);
251     }
252 },
253
```

```
254     showInfo() {
255         let token = cookie.get('token')
256         if (token) {
257             this.name = cookie.get('name')
258             console.log(this.name)
259         }
260     },
261
262     loginMenu(command) {
263         if('/logout' == command) {
264             cookie.set('name', '', {domain: 'localhost'})
265             cookie.set('token', '', {domain: 'localhost'})
266
267             //跳转页面
268             window.location.href = '/'
269         } else {
270             window.location.href = command
271         }
272     },
273
274     handleSelect(item) {
275         window.location.href = '/hospital/' + item.hoscode
276     },
277
278     weixinLogin() {
279         this.dialogAttr.showLoginType = 'weixin'
280     },
281
282     phoneLogin() {
283         this.dialogAttr.showLoginType = 'phone'
284         this.showLogin()
285     }
286 }
287 }
288 </script>
```