

4-33 实现用户页面数据加载

140.143.132.225:8000/project-1/doc-81

一、编写后端项目代码

在 `TbUserDao.xml` 中定义查询用户概要信息的SQL语句。

```
1. <select id="searchUserSummary" parameterType="int" resultType="HashMap">
2.     SELECT u.name, u.photo, d.dept_name AS deptName
3.     FROM tb_user u LEFT JOIN tb_dept d ON u.dept_id=d.id
4.     WHERE u.id=#{userId} AND u.status = 1
5. </select>
```

在 `TbUserDao.java` 接口中，定义抽象方法。

```
1. @Mapper
2. public interface TbUserDao {
3.     .....
4.     public HashMap searchUserSummary(int userId);
5. }
```

在 `UserService.java` 接口中，定义抽象方法。

```
1. public interface UserService {
2.     .....
3.     public HashMap searchUserSummary(int userId);
4. }
```

在 `UserServiceImpl.java` 中，定义实现方法。

```
1. public class UserServiceImpl implements UserService {
2.     .....
3.     @Override
4.     public HashMap searchUserSummary(int userId) {
5.         HashMap map = userDao.searchUserSummary(userId);
6.         return map;
7.     }
8. }
```

在 `UserController.java` 中，定义Web方法。

```
1. public class UserController {
2.     .....
3.     @GetMapping("/searchUserSummary")
4.     @ApiOperation("查询用户摘要信息")
5.     public R searchUserSummary(@RequestHeader("token") String token) {
6.         int userId = jwtUtil.getUserId(token);
7.         HashMap map = userService.searchUserSummary(userId);
8.         return R.ok().put("result", map);
9.     }
10. }
```

二、编写移动端代码

在 `main.js` 中声明全局请求路径

```
1. Vue.prototype.url = {
2.     .....
3.     searchUserSummary: baseUrl + "/user/searchUserSummary",
4. }
```

编写 `mine.vue` 页面中的 `onShow` 回调函数

```
1. onShow: function() {
2.     let that = this;
3.     that.ajax(that.url.searchUserSummary, "GET", null, function(resp){
4.         let result = resp.data.result;
5.         that.name = result.name;
6.         that.deptName = result.deptName;
7.         that.photo = result.photo;
8.     })
9. }
```

