

# Vue 网络请求与登录页面实现

主讲人：和凌志

# 主要内容

---

- 网络请求 (axios)
- 登录页面功能的实现

# 登录效果

对用户名和密码进行验证

☆

✔ 登录成功

我的管理系统

admin

.....

登录

---

# Vue 网络请求

## Vue 网络请求

- Vue 本身是不支持 ajax 调用的，如果需要这些功能就需要安装对应的工具；
- 支持 ajax 请求的工具很多，像 superagent 和 axios。这里，我们用到是 axios。

## axios 的安装

- `npm install axios --save`
- 在安装的时候，一定要切换进入到项目所在的根目录。再运行安装指令

## 在 main.js 文件中，引入 axios

JS main.js



src > JS main.js > ...

```
7   import App from './App'
8   import router from './router'
9
10  import axios from 'axios'
11  Vue.prototype.$http = axios;
12
13  Vue.config.productionTip = false
14
15  Vue.use(ElementUI);
16
```

- 在 main.js 文件中引入 axios，并加入到Vue 原型中；
- 在其他组件中，即可直接调用 Vue.\$http 或 this.\$http

```
import axios from 'axios'
Vue.prototype.$http = axios;
```

这是之前的 hardcode 模式，进一步修改为网络请求模式

```
methods: {  
  handleLogin() {  
    if ( ( this.form.username === 'admin' ) &&  
      ( this.form.password === '123456' ) ) {  
      Message({  
        message: "登录成功",  
        type: "success",  
        duration: 1 * 1000  
      });  
      this.$router.push("/manage");  
    } else {
```



# 登录网络请求的实现

```
login.vue ×
c > views > login.vue > {} "login.vue" > script > handleLogin
70 methods: {}
71 handleLogin() {
72   this.$http({
73     method: "get",
74     url: "http://localhost:3001/login"
75     // params: {
76     //   name: this.loginForm.username,
77     //   pwd: this.loginForm.password
78     // }
79   })
80   .then(res => {
81     console.log("登录后返回的数据 =", res);
82     if (
83       res.data.username === this.loginForm.username &&
84       res.data.password === this.loginForm.password
85     ) {
86       Message({
87         message: "登录成功",
88         type: "success",
89         duration: 3 * 1000
90       });
91     } else {
```

## ■ 注意事项：

- 一定要启动 JSON Server

本工程，启动 JSON Server 的方法：

\$ npm run json

- 查看 JSON Server 的 DB.json 文件数据

## 网络请求格式

---

```
this.$http( {  
  method: "get",  
  url: "http://localhost:3001/login"  
  // params: {  
  //   name: this.loginForm.username,  
  //   pwd: this.loginForm.password  
  // }  
})  
  .then(res => { })
```

## db.json

db.json > ...

```
1 {  
2   "login": {  
3     "username": "admin",  
4     "pwd": "123456"  
5   }  
6 }
```

---

# 动手实验 —— 用户登录接口验证

## 熟悉登录网络请求接口

1. 修改 db.json 文件中的 用户名和密码
2. 验证登录的账号、密码修改是否有效

# 启动 JSON Server

启动 JSON Server的 方法:

```
$ json-server --watch db.json --port 3000
```

<http://localhost:3000/login>

```
{  
  username: "admin",  
  pwd: "123456"  
}
```

## 运行结果

✕ 用户名或密码有误

我的管理系统

aa

.....

登录

## 网络请求注意事项

- 一定要注意 url 中端口号
- 先用浏览器验证
- 也可以用 postman 工具验证