

# Vue 首页的实现

主讲人：和凌志

# 主要内容

---

- 导航栏
- 路由
- 用户管理的构建

# 首页效果

← → ↺

localhost:8080/#/manage

☆

{ }

⚙

📱

🌐

应用 设置 my favorite

🏠 首页

+ 用户管理 ^

用户列表

+ 商品管理 ^

商品列表

+ 订单管理 ^

订单列表

⚙ 设置 ^

管理员设置

新增用户

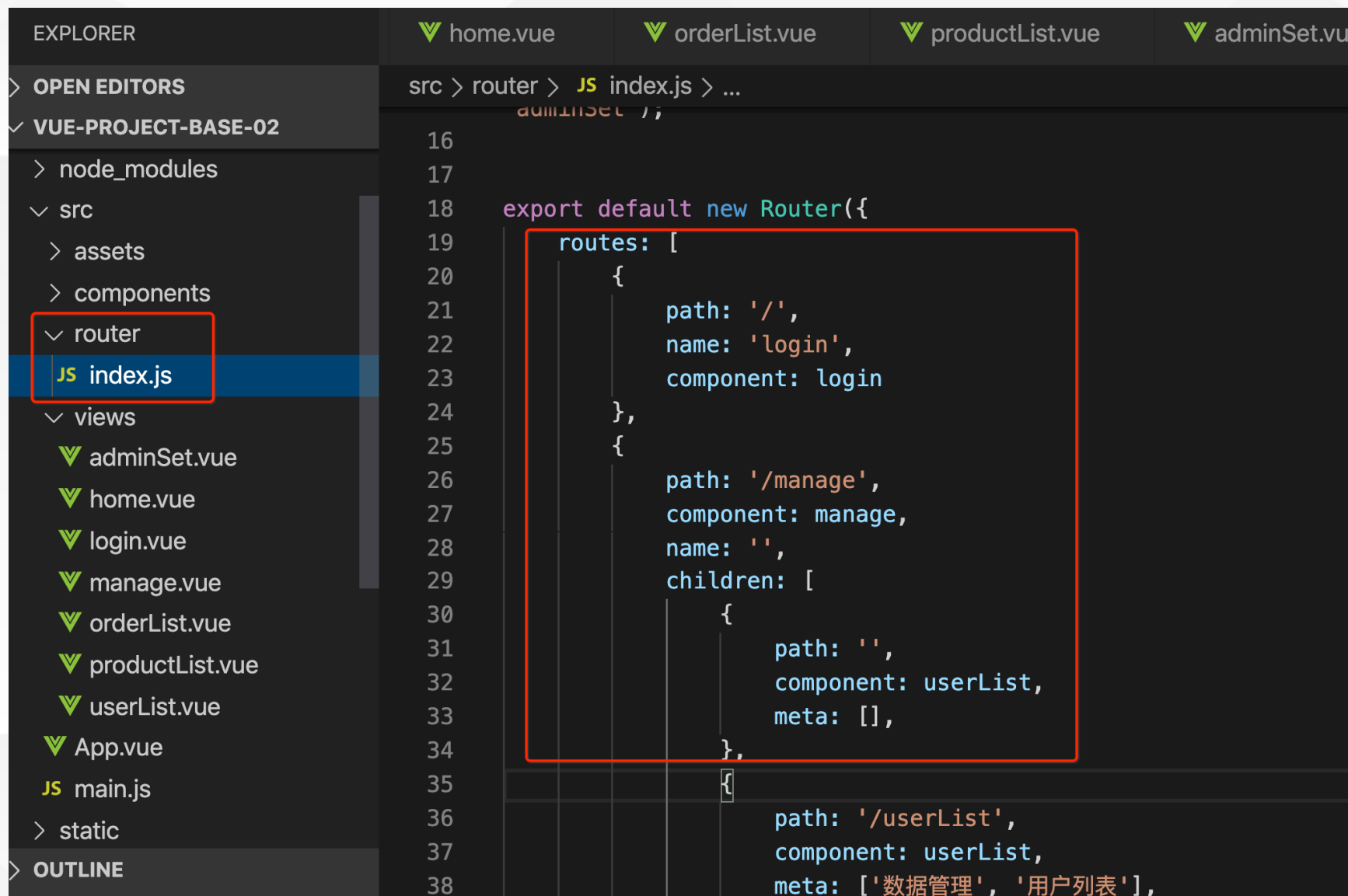
#	注册日期	用户姓名	注册地址	操作
1	2019-05-02	张三	北京市石景山区001	<div>编辑</div> <div>删除</div>
2	2019-05-03	李四	北京市石景山区	<div>编辑</div> <div>删除</div>
3	2019-05-02	张三	北京市石景山区001	<div>编辑</div> <div>删除</div>
4	2019-05-03	李四	北京市石景山区	<div>编辑</div> <div>删除</div>
5	2019-05-02	张三	北京市石景山区001	<div>编辑</div> <div>删除</div>
6	2019-05-03	李四	北京市石景山区	<div>编辑</div> <div>删除</div>
7	2019-05-02	张三	北京市石景山区001	<div>编辑</div> <div>删除</div>
8	2019-05-03	李四	北京市石景山区	<div>编辑</div> <div>删除</div>
9	2019-05-03	李四	北京市石景山区	<div>编辑</div> <div>删除</div>

## 构建思路

---

- 构建首页组件（manage.vue）
- 首页布局：< router-view > 用法
- 参考 Element UI，找到合适的导航栏
- 找到合适的 Table 组件
- 构建路由
- 为每个子菜单创建一个对应的组件
- 登录成功后，跳转到首页（manage）
- 构建 JSON Server 模拟数据，用于 Table 的展示

## 构建路由： router -> index.js



EXPLORER

- OPEN EDITORS
- VUE-PROJECT-BASE-02
  - node\_modules
  - src
    - assets
    - components
    - router
      - JS index.js**
    - views
      - adminSet.vue
      - home.vue
      - login.vue
      - manage.vue
      - orderList.vue
      - productList.vue
      - userList.vue
    - App.vue
    - JS main.js
    - static
- OUTLINE

src > router > JS index.js > ...

```
16
17
18 export default new Router({
19   routes: [
20     {
21       path: '/',
22       name: 'login',
23       component: login
24     },
25     {
26       path: '/manage',
27       component: manage,
28       name: '',
29       children: [
30         {
31           path: '',
32           component: userList,
33           meta: [],
34         },
35         {
36           path: '/userList',
37           component: userList,
38           meta: ['数据管理', '用户列表'],
```

通过 lazy-loading 技术引入组件 ，

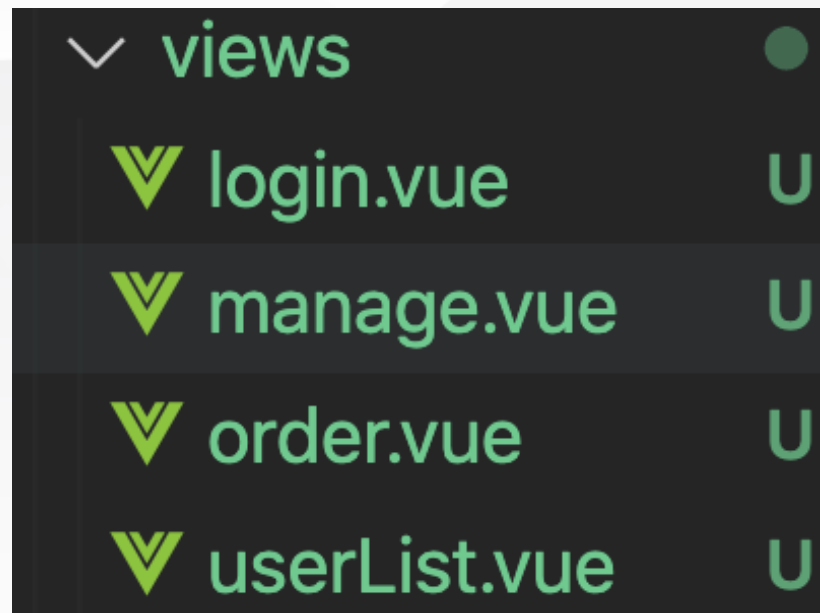
在 router -> index.js 文件中，编写代码。如下。Copy 时，注意单引号中英文

```
const userList = r => require.ensure([], () => r(require('@views/userList')), 'userList');
```

```
import login from '@views/login'  
import manage from '@views/manage'  
import order from '@views/order'
```

```
const userList = r => require.ensure([], () => r(require('@views/userList'))
```

## 创建每个导航条所需要的组件



- 组件的内容不做特别要求
- 预先准备好一个 template.vue

# 构建 userList 的 template

OPEN EDITORS

VUE-PROJECT-BASE-02

assets

components

router

index.js

views

adminSet.vue

home.vue

login.vue

manage.vue

orderList.vue

productList.vue

userList.vue

App.vue

main.js

static

.babelrc

.editorconfig

src > views > userList.vue > {} "userList.vue" > template > div.fillcontain

1 <template>

2 <div class="fillcontain">

3 <el-button type="primary" style="margin:20px">新增用户</el-button>

4 <div class="table\_container">

5 <el-table :data="tableData" highlight-current-row style="width: 100%">

6 <el-table-column type="index" width="100"></el-table-column>

7 <el-table-column property="register\_time" label="注册日期" width="220"></el-table-column>

8 <el-table-column property="username" label="用户姓名" width="220"></el-table-column>

9 <el-table-column property="city" label="注册地址"></el-table-column>

10 <el-table-column label="操作" width="160">

11 <template slot-scope="scope">

12 <el-button size="small" @click="handleEdit(scope.row)">编辑</el-button>

13 <el-button size="small" type="danger" @click="handleDelete(scope.\$index, scope.row)">删除</el-button>

14 </template>

15 </el-table-column>

16 </el-table>

17 </div>

18 </div>

19 </template>



## 构建 userList 的 Script

- 创建页面前，调用 JSON Server 的用户数据

```
this.$http({  
  method: "get",  
  url: "http://localhost:3001/users"  
})
```

注意：端口号与服务端一致

```
<script>  
export default {  
>   data() { ...  
     },  
   components: {},  
   created() {  
     this.initData();  
   },  
   methods: {  
     initData() {  
       this.$http({  
         method: "get",  
         url: "http://localhost:3001/users"  
       })  
>       .then(res => { ...  
         })  
       .catch(error => {  
         console.log(error);  
       });  
     }  
   }  
};  
</script>
```

## Table 数据映射关系：3个地方

```
<div class="table_container">
  <el-table :data="tableData" highlight-current-row style="width: 100%">
    <el-table-column type="index" width="100"></el-table-column>
    <el-table-column property="register_time" label="注册日期" width="220">
      <el-table-column property="username" label="用户名" width="220"></el-table-column>
    </el-table-column>
  </el-table>
</div>
```

```
export default {
  data() {
    return {
      tableData: []
    };
  },
}
```

```
methods: {
  initData() {
    this.$http({
      method: "get",
      url: "http://localhost:3000/users"
    })
      .then(res => {
        console.log(" 用户列表 =", res);
        this.tableData = res.data;
        this.count = res.data.length;
      })
      .catch(error => {
        console.log(error);
      });
  }
}
```

## 构建 JSON Server 数据

```
{
  "login": {
    "username": "admin",
    "password": "123456"
  },
  "users": [
    {
      "register_time": "2019-05-02",
      "username": "张三",
      "city": "北京市石景山区001"
    },
    {
      "register_time": "2019-05-03",
      "username": "李四",
      "city": "北京市石景山区"
    },
    {
      "register_time": "2019-05-02",
      "username": "张三",
      "city": "北京市石景山区001"
    }
  ]
}
```



# 动手实验

## 运行工程

- 运行当前的工程
- 期望：登录成功后，跳转到首页

## 扩展练习 - 添加一个 商品列表

### 从后台 到前端：

- 构建后台接口 <http://localhost:3000/products>
- 构建后台数据， 修改 db.json 文件
- 验证后台接口， 通过浏览器验证

### 前端：

- 添加商品列表组件
- 添加路由
- 添加网络请求

## 后台数据的修改

- 在 db.json 中修改数据 （注意：修改数据库后，重新启动 JSON Server）

# 下次课

---

✓ 实现 增、删、改、查