

## 0. 实现步骤

---

1. 安装并配置 vue-router 4.x
2. 展示 Login.vue 登录组件
3. 模拟并实现登录功能
4. 通过路由渲染 Home.vue
5. 实现退出登录的功能
6. 全局控制路由的访问权限
7. 将左侧菜单改造为路由链接
8. 渲染用户管理页面的数据
9. 实现跳转到用户详情页的功能
10. 开启路由的 props 传参
11. 通过程式导航实现后退功能

## 1. 安装并配置 vue-router 4.x

---


1. 运行如下的命令，安装 `vue-router`：

```
1 npm install vue-router@next -S
```

2. 在 `src` 目录下新建 `router.js` 路由模块：

```
1 // 1. 按需导入对应的函数
2 import { createRouter, createWebHashHistory } from 'vue-router'
3
4 // 2. 创建路由对象
5 const router = createRouter({
6   history: createWebHashHistory(),
7   routes: [],
8 })
9
10 // 3. 向外共享路由实例对象
11 export default router
```

3. 在 `main.js` 入口文件中导入并挂载路由对象：



```
1 // 1. 导入路由模块
2 import router from './router'
3
4 const app = createApp(App)
5
6 // 2. 挂载路由对象
7 app.use(router)
8
9 app.mount('#app')
```

## 2. 展示 Login.vue 登录组件

---

1. 在 `router.js` 模块中导入 `Login.vue` 组件:



```
1 import Login from './components/MyLogin.vue'
```

2. 声明路由规则如下:



```
1 routes: [
2   // 路由重定向
3   { path: '/', redirect: '/login' },
4   { path: '/login', component: Login },
5 ]
```

3. 在 `App.vue` 组件中声明路由占位符:



```
1 <template>
2   <!-- 路由的占位符 -->
3   <router-view></router-view>
4 </template>
5
6 <script>
7   export default {
8     name: 'MyApp',
9   }
10 </script>
11
12 <style lang="less" scoped>
13 </style>
```

### 3. 模拟并实现登录功能

1. 在 `MyLogin.vue` 组件中声明如下的 `data` 数据：

```
1 data() {  
2     return {  
3         username: '',  
4         password: '',  
5     }  
6 },
```

2. 为 **用户名** 和 **密码** 的文本框进行 `v-model` 双向数据绑定：

```
1 <!-- 登录名称 -->  
2 <div class="form-group form-inline">  
3     <label for="username">登录名称</label>  
4     <input type="text" class="form-control ml-2" id="username"  
5         placeholder="请输入登录名称" autocomplete="off" v-model="username">  
6 </div>  
7 <!-- 登录密码 -->  
8 <div class="form-group form-inline">  
9     <label for="password">登录密码</label>  
10    <input type="password" class="form-control ml-2" id="password"  
11        placeholder="请输入登录密码" v-model="password">  
12 </div>
```

3. 为 **登录按钮** 绑定点击事件处理函数：

```
1 <button type="button" class="btn btn-primary"  
2     @click="onLoginClick">登录</button>
```

4. 在 `methods` 中声明 `onLoginClick` 事件处理函数如下：

```

1  methods: {
2    onLoginClick() {
3      // 判断用户名和密码是否正确
4      if (this.username === 'admin' && this.password === '123456') {
5        // 登录成功，跳转到后台主页
6        this.$router.push('/home')
7        // 模拟存储 Token 的操作
8        return localStorage.setItem('token', 'Bearer xxx')
9      }
10     // 登录失败，清除 Token
11     localStorage.removeItem('token')
12   },
13 },

```

## 4. 通过路由渲染 Home.vue

1. 在 `router.js` 中导入 `Home.vue` 组件:

```

1  import Home from './components/MyHome.vue'

```

2. 在 `routes` 路由规则的数组中，声明对应的路由规则:

```

1  routes: [
2    { path: '/', redirect: '/login' },
3    { path: '/login', component: Login },
4    // Home 组件的路由规则
5    { path: '/home', component: Home },
6  ]

```

3. 渲染 `Home.vue` 组件的基本结构:

```

1  <template>
2    <div class="home-container">
3      <!-- 头部组件 -->
4      <my-header></my-header>
5
6      <!-- 主体区域 -->
7      <div class="home-main-box">
8

```

```
9      <!-- 左侧边栏区域 -->
10      <my-aside></my-aside>
11
12      <!-- 右侧内容主体区域 -->
13      <div class="home-main-body"></div>
14
15      </div>
16    </div>
17  </template>
```

## 5. 实现退出登录的功能

1. 在 `MyHeader.vue` 组件中, 为 `退出登录` 按钮绑定 `click` 事件处理函数:

```
1 <button type="button" class="btn btn-light" @click="onLogout">退出
  登录</button>
```

2. 在 `methods` 中声明如下的事件处理函数:

```
1 export default {
2   name: 'MyHeader',
3   methods: {
4     // 退出按钮的点击事件处理函数
5     onLogout() {
6       // 移除 Token
7       localStorage.removeItem('token')
8       // 强制跳转到“登录页面”
9       this.$router.push('/login')
10    },
11  },
12 }
```

## 6. 全局控制路由的访问权限

1. 在 `router.js` 模块中, 通过 `router` 路由实例对象, 全局挂载路由导航守卫:

```
1 // 全局路由导航守卫
2 router.beforeEach((to, from, next) => {
3   // 如果用户访问的是登录页面，直接放行
4   if (to.path === '/login') return next()
5   // 获取 Token 值
6   const token = localStorage.getItem('token')
7   if (!token) {
8     // Token 值不存在，强制跳转到登录页面
9     return next('/login')
10  }
11  // 存在 Token 值，直接放行
12  next()
13 })
```

## 7. 将左侧菜单改造为路由链接

1. 打开 `MyAside.vue` 组件，把 `li` 内部的纯文本升级改造为 `<router-link>` 组件：

```
1 <template>
2   <div class="layout-aside-container">
3     <!-- 左侧边栏列表 -->
4     <ul class="user-select-none menu">
5       <li class="menu-item">
6         <router-link to="/home/users">用户管理</router-link>
7       </li>
8       <li class="menu-item">
9         <router-link to="/home/rights">权限管理</router-link>
10      </li>
11      <li class="menu-item">
12        <router-link to="/home/goods">商品管理</router-link>
13      </li>
14      <li class="menu-item">
15        <router-link to="/home/orders">订单管理</router-link>
16      </li>
17      <li class="menu-item">
18        <router-link to="/home/settings">系统设置</router-link>
19      </li>
20    </ul>
21  </div>
22 </template>
```

2. 打开 `Home.vue` 组件, 在 **右侧内容主体区域** 中声明子路由的占位符:

```
1 <template>
2   <div class="home-container">
3     <!-- 头部组件 -->
4     <my-header></my-header>
5     <!-- 主体区域 -->
6     <div class="home-main-box">
7       <!-- 左侧边栏区域 -->
8       <my-aside></my-aside>
9       <!-- 右侧内容主体区域 -->
10      <div class="home-main-body">
11        <!-- **子路由的占位符** -->
12        <router-view></router-view>
13      </div>
14    </div>
15  </div>
16 </template>
```

3. 在 `router.js` 中导入左侧菜单对应的组件:

```
1 import Users from './components/menus/MyUsers.vue'
2 import Rights from './components/menus/MyRights.vue'
3 import Goods from './components/menus/MyGoods.vue'
4 import Orders from './components/menus/MyOrders.vue'
5 import Settings from './components/menus/MySettings.vue'
```

4. 通过 `children` 属性, 为 `home` 规则定义子路由规则如下:

```
1 {
2   path: '/home',
3   component: Home,
4   // 用户访问 /home 时, 重定向到 /home/users
5   redirect: '/home/users',
6   // 子路由规则
7   children: [
8     { path: 'users', component: Users },
9     { path: 'rights', component: Rights },
10    { path: 'goods', component: Goods },
11    { path: 'orders', component: Orders },
12    { path: 'settings', component: Settings },
13  ],
14 },
```

## 8. 渲染用户管理页面的数据

1. 在 `MyUsers.vue` 组件中, 通过 `v-for` 指令循环渲染用户列表的数据:

```
1 <tbody>
2   <tr v-for="(item, i) in userlist" :key="item.id">
3     <td>{{i + 1}}</td>
4     <td>{{item.name}}</td>
5     <td>{{item.age}}</td>
6     <td>{{item.position}}</td>
7     <td>详情</td>
8   </tr>
9 </tbody>
```

## 9. 实现跳转到用户详情页的功能

1. 在 `MyUsers.vue` 组件中, 渲染详情页的**路由链接**如下:

```
1 <td>
2   <router-link :to="'/home/users/' + item.id">详情</router-link>
3 </td>
```

2. 在 `router.js` 中导入**用户详情页**组件:

```
1 import UserDetails from './components/user/MyUserDetail.vue'
```

3. 在 `home` 规则的 `children` 节点下, 声明 **用户详情页** 的路由规则:

```
1 {
2   path: '/home',
3   component: Home,
4   redirect: '/home/users',
5   children: [
6     { path: 'users', component: Users },
7     { path: 'rights', component: Rights },
8     { path: 'goods', component: Goods },
9     { path: 'orders', component: Orders },
```



```

10     { path: 'settings', component: Settings },
11     // 用户详情页的路由规则
12     { path: 'users/:id', component: UserDetails },
13   ],
14 },

```

## 10. 开启路由的 props 传参

1. 在 `router.js` 模块中, 为 用户详情页 的路由规则开启 `props` 传参:

```

1 { path: 'users/:id', component: UserDetails, props: true },

```

2. 在 `MyUserDetail.vue` 组件中声明 `props` 参数:

```

1 export default {
2   name: 'MyUserDetail',
3   props: ['id'],
4 }

```

3. 在 `MyUserDetail.vue` 组件的结构中直接使用路由参数:

```

1 <template>
2   <button type="button" class="btn btn-light btn-sm">后退</button>
3   <h4 class="text-center">用户详情 --- {{id}}</h4>
4 </template>

```

## 11. 通过程式化导航实现后退功能

1. 在 `MyUserDetail.vue` 组件中, 为 后退按钮 绑定点击事件处理函数:

```

1 <template>
2   <button type="button" class="btn btn-light btn-sm"
3     @click="goBack">后退</button>
4   <h4 class="text-center">用户详情 --- {{id}}</h4>
5 </template>

```

2. 在 `methods` 中声明 `goBack` 事件处理函数如下:



```
1 export default {
2   name: 'MyUserDetail',
3   props: ['id'],
4   methods: {
5     // 编程式导航实现后退功能
6     goBack() {
7       this.$router.go(-1)
8     },
9   },
10 }
```