

0. 实现步骤

1. 初始化项目
2. 渲染用户列表组件
3. 基于全局过滤器处理时间格式
4. 实现添加用户的操作
5. 实现删除用户的操作
6. 通过路由跳转到详情页

1. 初始化项目

1.1 梳理项目结构

1. 基于 `vue-cli` 运行如下的命令，新建 `vue2.x` 的项目：

```
1 vue create code-users
```

2. 重置 `App.vue` 组件中的代码：

```
1 <template>
2   <div>
3     <h1>App 组件</h1>
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: 'MyApp',
10  }
11 </script>
12
13 <style>
14 </style>
```

3. 删除 `components` 目录下的 `HelloWorld.vue` 组件。

1.2 修改项目的开发调试配置

1. 在项目根目录中新建 `vue.config.js` 配置文件。

2. 在 `vue.config.js` 配置文件中, 通过 `devServer` 节点添加如下的配置项:

```
1 module.exports = {
2   devServer: {
3     // 修改 dev 期间的端口号
4     port: 3000,
5     // 自动打开浏览器
6     open: true
7   }
8 }
```

1.3 初始化路由

1. 运行如下的命令, 在 `vue2.x` 的项目中安装 `vue-router`:

```
1 npm install vue-router@3.4.9 -S
```

2. 在 `src` 目录下新建 `router/index.js` 路由模块:

```
1 // 路由模块
2 import Vue from 'vue'
3 import VueRouter from 'vue-router'
4
5 // 安装路由插件
6 Vue.use(VueRouter)
7
8 // 创建路由实例对象
9 const router = new VueRouter({
10   // 路由规则
11   routes: [],
12 })
13
14 // 向外共享路由实例对象
15 export default router
```

3. 在 `main.js` 模块中导入并挂载路由模块:

```
1 import Vue from 'vue'
2 import App from './App.vue'
3 // 导入路由模块
4 import router from './router'
5
6 Vue.config.productionTip = false
7
8 new Vue({
9   // 挂载路由
10   router,
11   render: h => h(App),
12 }).$mount('#app')
```

1.4 使用路由渲染 UserList 组件

1. 在 `components` 目录下新建 `UserList.vue` 组件如下:

```
1 <template>
2   <div>
3     UserList
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: 'UserList',
10   }
11 </script>
12
13 <style>
14 </style>
```

2. 在 `router/index.js` 路由模块中新增如下的路由规则:

```
1 // 创建路由实例对象
2 const router = new VueRouter({
3   // 路由规则
4   routes: [
5     // 路由重定向
6     { path: '/', redirect: '/users', },
7     // 用户列表的路由规则
8     { path: '/users', component: UserList }
9   ],
10 })
```

3. 在 `App.vue` 中声明 `router-view` 路由占位符：

```
1 <template>
2   <div>
3     <!-- 路由占位符 -->
4     <router-view></router-view>
5   </div>
6 </template>
7
8 <script>
9   export default {
10     name: 'MyApp',
11   }
12 </script>
```

2. 渲染用户列表组件

2.1 安装并配置 axios

1. 运行如下的命令，在项目中安装 `axios`：

```
1 npm install axios@0.21.1 -S
```

2. 在 `main.js` 中导入并配置 `axios`：

```
1 import Vue from 'vue'
2 import App from './App.vue'
3 import router from './router'
```

```
4 // 导入 axios
5 import axios from 'axios'
6
7 Vue.config.productionTip = false
8
9 // 全局配置 axios
10 axios.defaults.baseURL = 'https://www.escook.cn'
11 Vue.prototype.$http = axios
12
13 new Vue({
14   router,
15   render: h => h(App),
16 }).$mount('#app')
```

2.2 请求用户列表的数据

1. 在 `UserList.vue` 组件中声明如下的 `data` 数据节点:

```
1 data() {
2   return {
3     // 用户列表数据，默认为空数组
4     userList: [],
5   }
6 }
```

2. 在 `created` 生命周期函数中预调用 `getUserList` 方法:

```
1 created() {
2   // 调用此方法，请求用户列表数据
3   this.getUserList()
4 }
```

3. 在 `methods` 中声明 `getUserList` 方法:

```

1  methods: {
2    // 请求用户列表的数据
3    async getUserList() {
4      const { data: res } = await this.$http.get('/api/users')
5      // res.status 等于 0 表示数据请求成功，否则，请求失败！
6      if (res.status !== 0) return console.log('用户列表数据请求失败！')
7      this.userList = res.data
8    },
9  }

```

2.3 解决跨域请求限制

由于 API 接口服务器并没有开启 CORS 跨域资源共享，因此终端会提示如下的错误：

Access to XMLHttpRequest at 'https://www.escook.cn/api/users' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

解决方案：

通过 `vue.config.js` 中的 `devServer.proxy` 即可在**开发环境下**将 API 请求**代理到** API 服务器。

```

1  module.exports = {
2    devServer: {
3      port: 3000,
4      open: true,
5      // 当前项目在开发调试阶段，
6      // 会将任何未知请求（没有匹配到静态文件的请求）代理到
7      // https://www.escook.cn
8      proxy: 'https://www.escook.cn'
9    }
10 }

```

同时，在 `main.js` 入口文件中，需要把 `axios` 的**根路径**改造为**开发服务器的根路径**：

```

1  // 全局配置 axios
2  Vue.prototype.$http = axios
3  axios.defaults.baseURL = 'http://localhost:3000'

```

注意：devServer.proxy 提供的代理功能，仅在开发调试阶段生效。项目上线发布时，依旧需要 API 接口服务器开启 CORS 跨域资源共享。

2.4 安装并配置 element-ui

1. 运行如下的命令，在项目中安装 element-ui 组件库：

```
1 npm i element-ui -S
```

2. 在 `main.js` 中配置 element-ui：

```
1 import Vue from 'vue'
2 import App from './App.vue'
3 import router from './router'
4 import axios from 'axios'
5 // 1. 导入 element-ui
6 import ElementUI from 'element-ui'
7 // 2. 导入 element-ui 的样式表
8 import 'element-ui/lib/theme-chalk/index.css'
9
10 Vue.config.productionTip = false
11 // 3. 将 ElementUI 安装为 vue 的插件
12 Vue.use(ElementUI)
13
14 Vue.prototype.$http = axios
15 axios.defaults.baseURL = 'http://localhost:3000'
16
17 new Vue({
18   router,
19   render: h => h(App),
20 }).$mount('#app')
```

3. 项目中用到的 API 接口

3.1 请求根路径

<https://www.escook.cn/>

3.2 获取用户列表

- 请求方式：
 - GET

- 请求地址：
 - /api/users
- 请求参数：
 - 无

3.3 添加用户

- 请求方式：
 - POST
- 请求地址：
 - /api/users
- 请求参数：
 - name 用户姓名 (1 - 15 个字符之间)
 - age 用户年龄 (1 - 100 之间)
 - position 职位 (1 - 10 个字符之间)
- 请求结果：
 - status 的值等于 0 表示成功

3.4 删除用户

- 请求方式：
 - delete
- 请求地址：
 - /api/users/:id
- 请求参数：
 - id 要删除的用户的Id (URL参数)
- 请求结果：
 - status 的值等于 0 表示成功

3.5 获取用户信息

- 请求方式：
 - GET
- 请求地址：
 - /api/users/:id
- 请求参数：
 - id 要查询的用户的Id (URL参数)
- 请求结果：
 - status 的值等于 0 表示成功