

# 前端登陆功能

## 目录

前端登陆功能 .....	1
目标任务 .....	2
思路分析 .....	2
● 登录页面的展示 .....	2
编码步骤 .....	4
● 效果图 .....	4
● 最终目录结构 .....	7
● 编程实现 .....	8
1. 第三方组件安装 .....	9
2. 第三方组件其他资料 .....	9
3. vue-router 基本使用 .....	9
4. 使用 Vue-router 实现登录页面跳转 .....	11
5. 使用 element-ui 完成登录页面布局 .....	14
6. 使用 Axios 向后端服务器发送请求进行登录 .....	21
7. 使用 Vuex 管理登录状态 .....	25
源码参考 .....	34
回马枪总结 .....	34

## 目标任务

- 登录页面的展示
- 用户名密码的判断
- 登录成功页面的跳转

## 相关知识点

- Element 的 Button、Input 组件的使用
- Router 和子 Router 的使用
- Vue 的 Class 绑定
- Vue 的事件处理
- Vue 的 router 的使用

## 思路分析

- 登录页面的展示

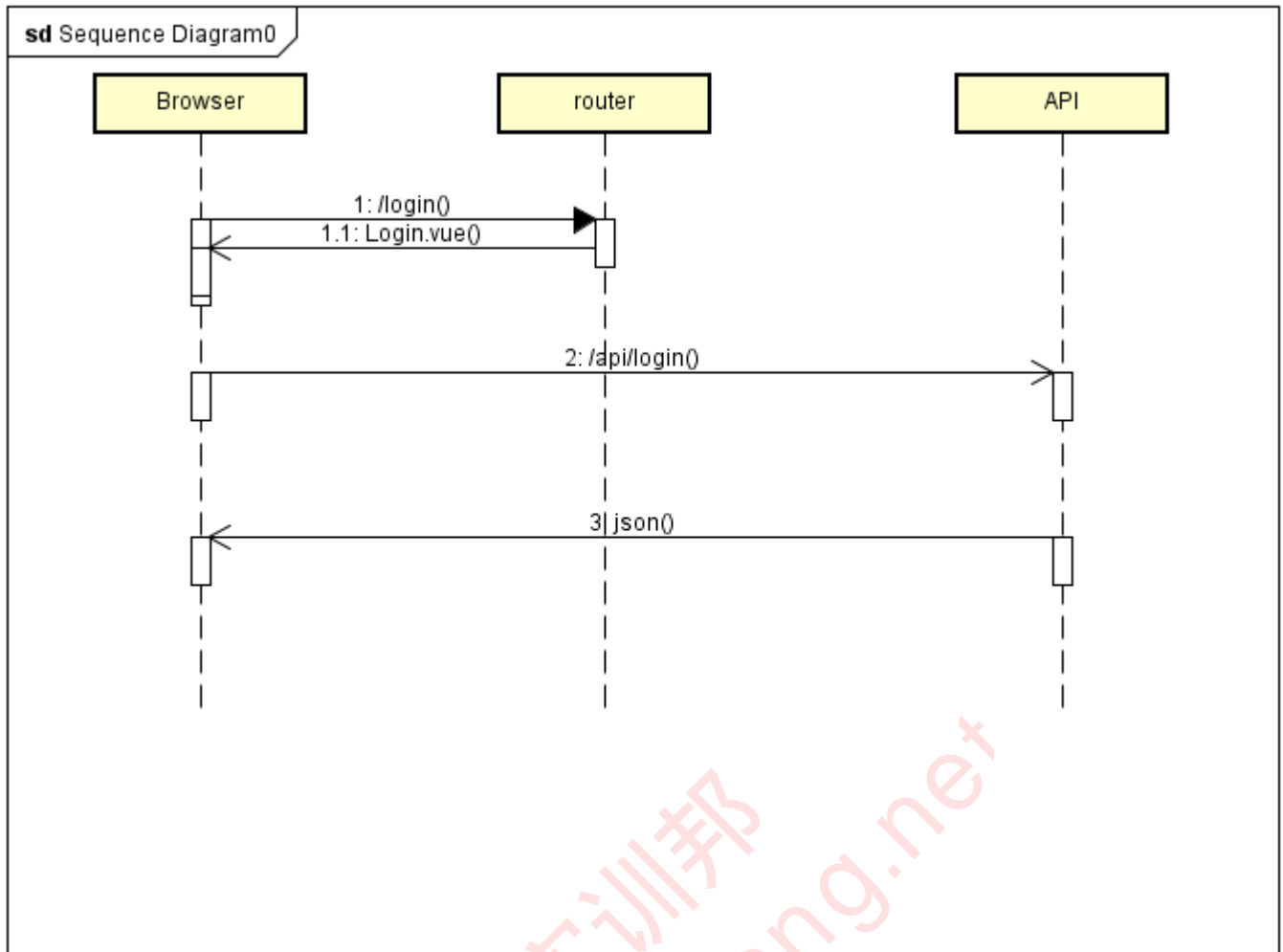
后台 API:

/api/login

前台 router:

/login

功能流程图:



API 接口:

**url:/api/login**

method: post

逻辑:

1. 进入页面，展示基本元素

/login 进入 Login.vue

2. 异步请求数据

通过 Login.vue 页面发起请求 (/api/login) ，并接收 json 数据

3. 渲染数据或者提示信息

处理返回的 json 数据进行页面的数据渲染

4. 跳转到/home/courselist 页面

If(后台响应返回:成功){

```
if(用户名、密码正确) {
```

```
    登录成功
```

```
    修改登录状态
```

```
    跳转到/home/courselist
```

```
}else{
```

```
    登录失败，用户名或密码错误
```

```
}else{
```

```
    系统出错
```

```
}
```

参数：

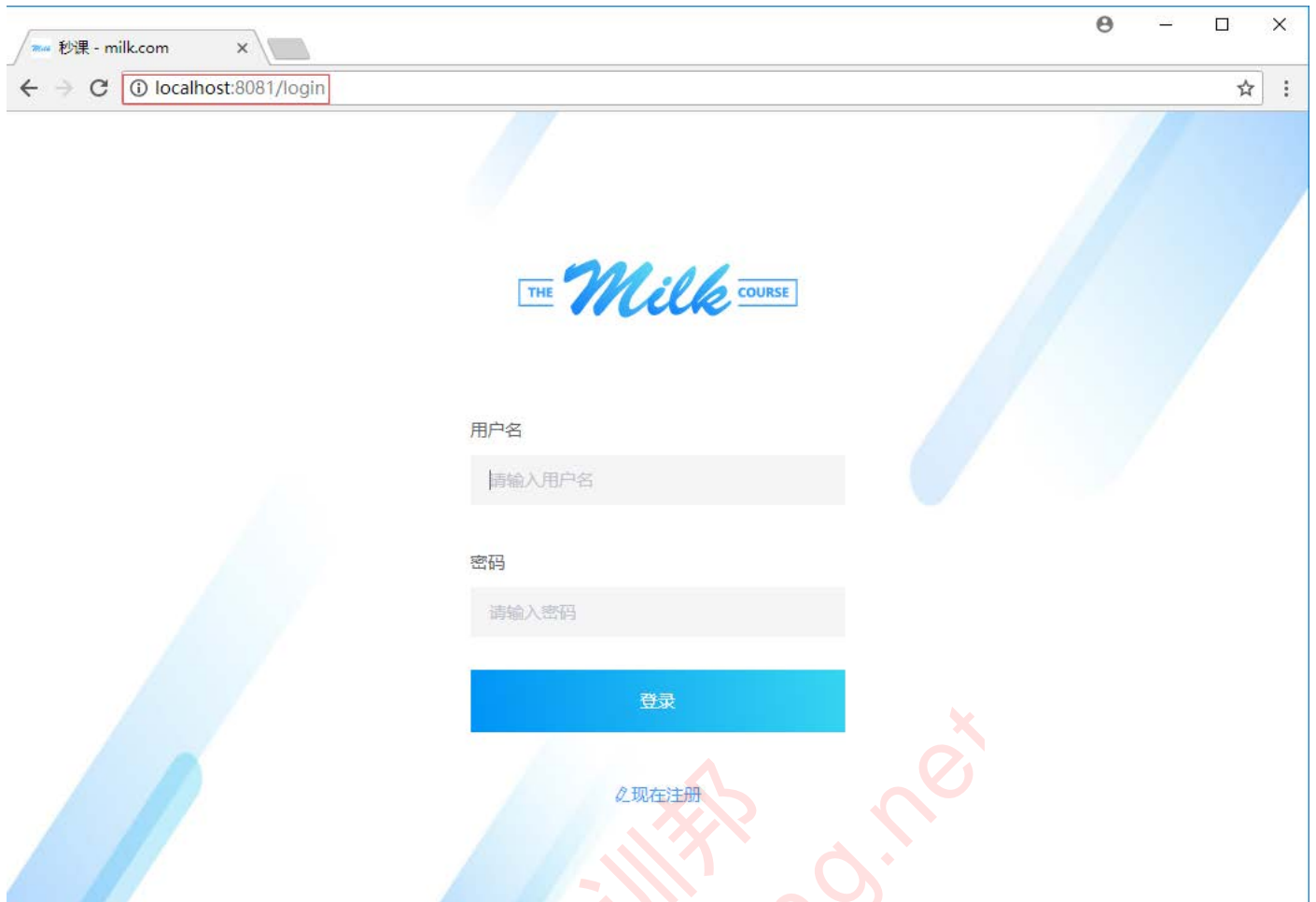
参数名	类型	是否必须	说明
username	String	Y	用户名
password	String	Y	密码

返回值：

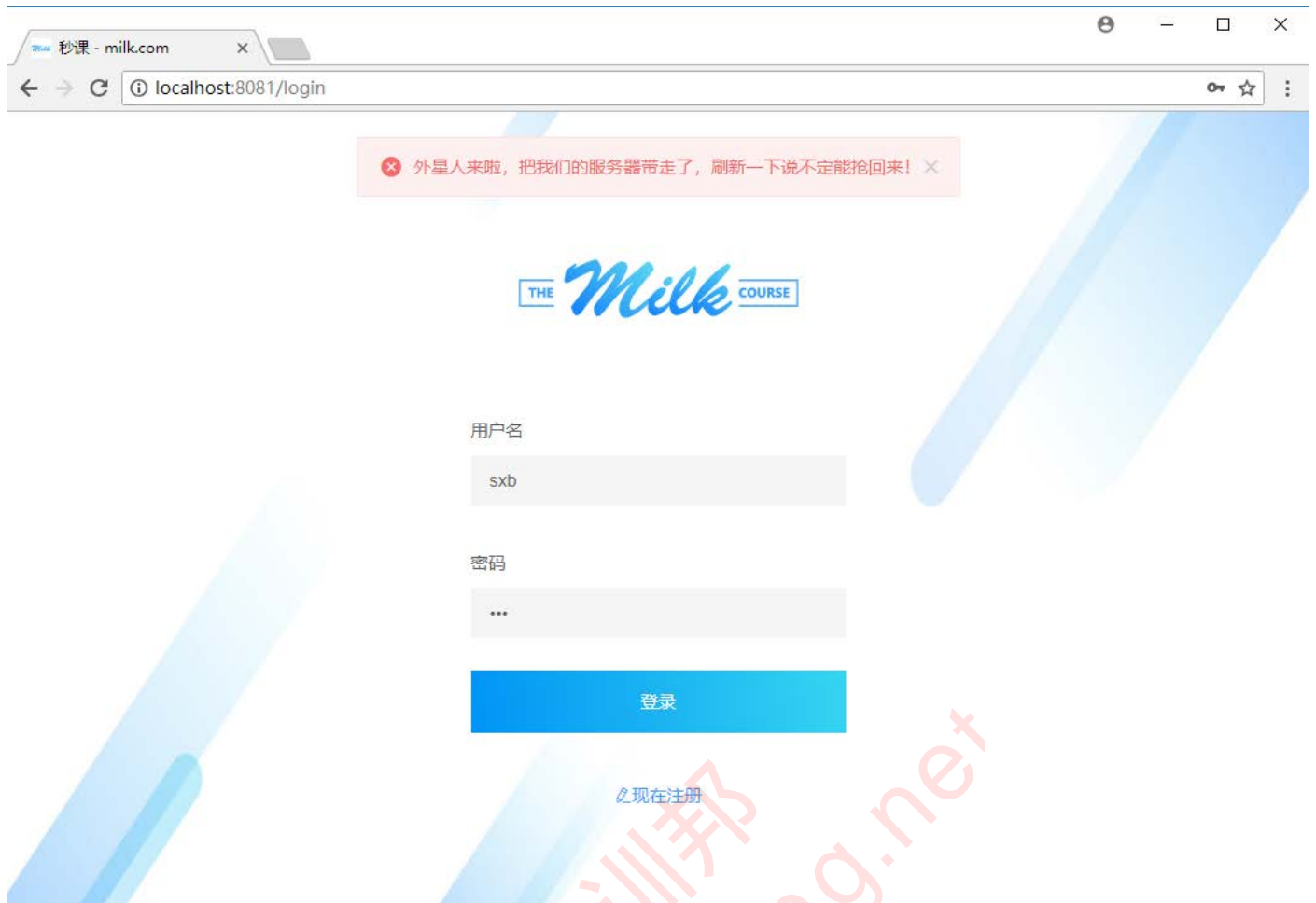
参数名	类型	是否必须	说明
code	Int	Y	200 为成功 500***为失败
message	String	Y	成功或失败的消息内容
data	object	N	

## 编码步骤

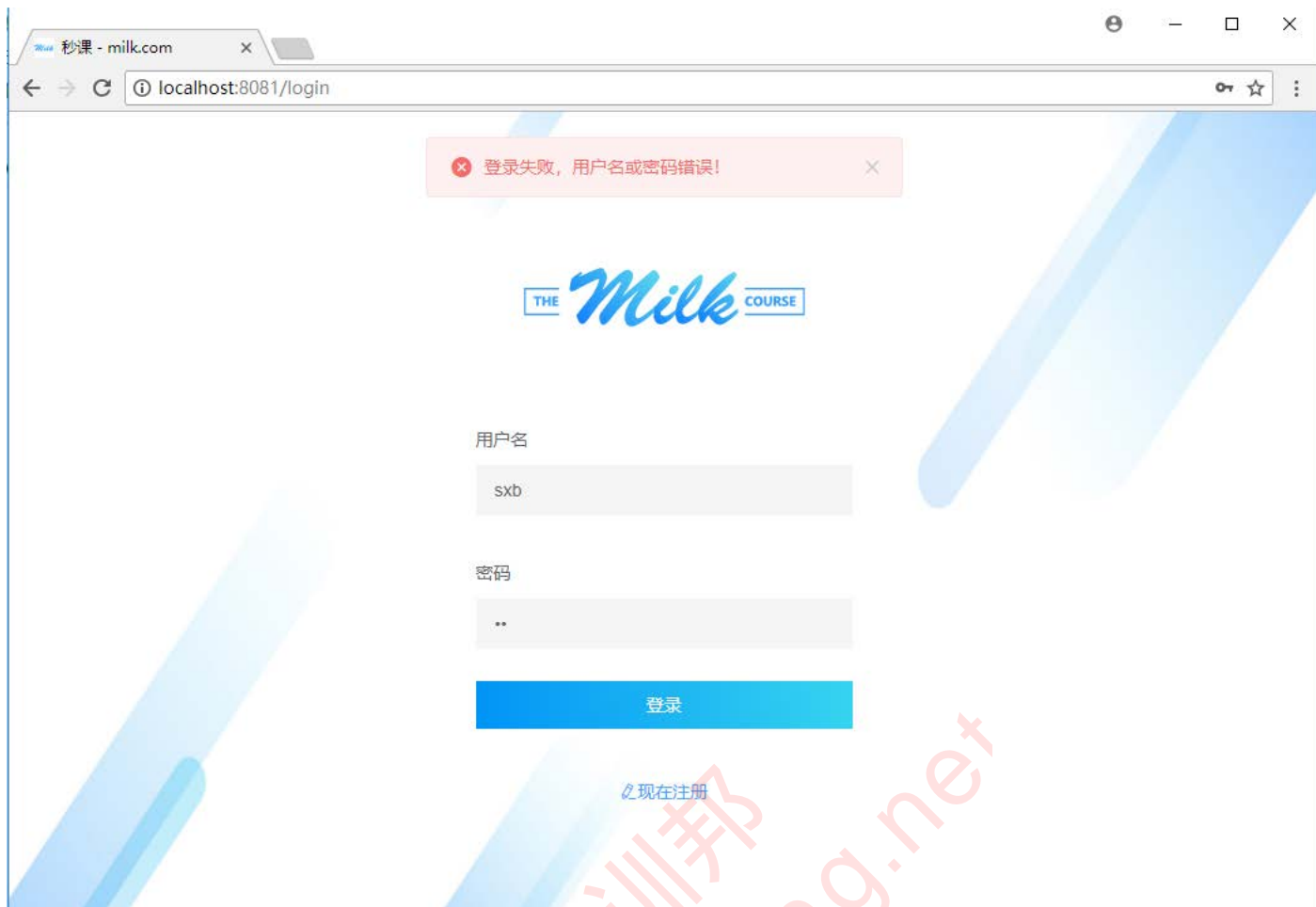
- 效果图



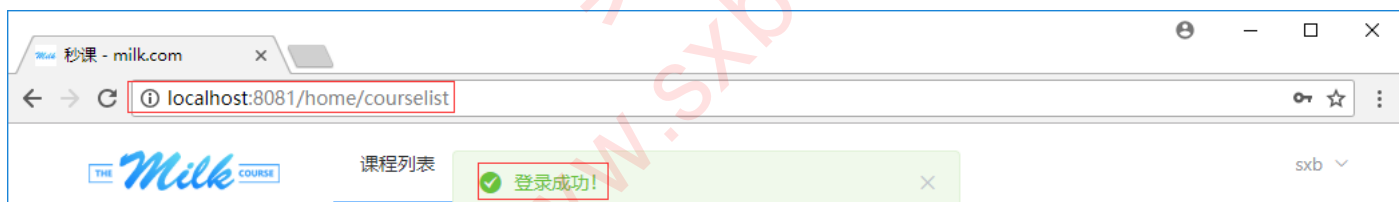
没有启动后端项目，提示登录失败



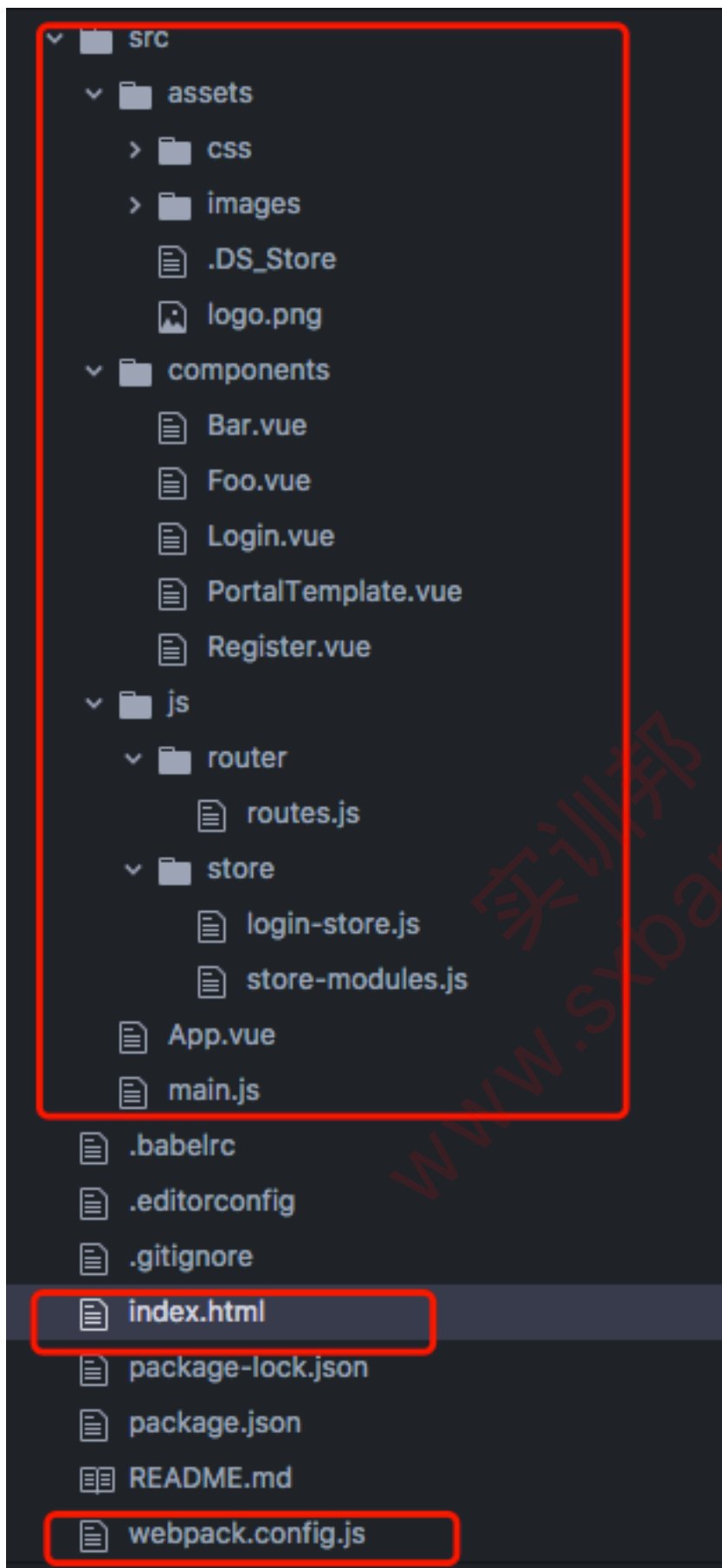
密码输入错误, 提示登录失败



启动后端项目并且用户名和密码都填写正确跳转到课程列表并提示登录成功



- 最终目录结构





## 1. 第三方组件安装

- vue-router: npm install vue-router -d
- element-ui: npm install element-ui -d
- axios: npm install axios -d
- vue-axios: npm install vue-axios -d
- qs: npm install qs -d
- vuex: npm install vuex -d

## 2. 第三方组件其他资料

- vue-router: <https://router.vuejs.org/zh/guide/#html>
- element-ui: <http://element-cn.eleme.io/#/zh-CN/component/installation>
- axios: <https://github.com/axios/axios>
- vue-axios: <https://www.npmjs.com/package/vue-axios>
- qs: <https://www.npmjs.com/package/qs>
- vuex: <https://vuex.vuejs.org/zh/>

## 3. vue-router 基本使用

- App.vue <router-link>

```
<template>
<div id="app">
  <h1>Hello App!</h1>
  <p>
    <!-- 使用 router-link 组件来导航。 -->
    <!-- 通过传入 `to` 属性指定链接。 -->
    <!-- <router-link> 默认会被渲染成一个 `<a>` 标签 -->
    <router-link to="/foo">Go to Foo</router-link>
    <router-link to="/bar">Go to Bar</router-link>

    <!-- 需要一些其他业务时 -->
    <button @click="gotoByUrl('/foo')">Go to Foo</button>
    <button @click="gotoByUrl('/bar')">Go to Bar</button>
  </p>
</div>
```

```

<!-- 路由出口 -->
<!-- 路由匹配到的组件将渲染在这里 -->
<router-view></router-view>
</div>
</template>

<script>
export default {
  methods: {
    gotoByUrl(url) {
      // do some things
      this.$router.push(url)
    }
  }
}
</script>

```

#### ● main.js

```

// 0. 如果使用模块化机制编程，导入 Vue 和 VueRouter，要调用 Vue.use(VueRouter)

// 1. 定义（路由）组件。
// 可以从其他文件 import 进来
const Foo = { template: '<div>foo</div>' }
const Bar = { template: '<div>bar</div>' }

// 2. 定义路由
// 每个路由应该映射一个组件。 其中"component" 可以是
// 通过 Vue.extend() 创建的组件构造器，
// 或者，只是一个组件配置对象。
// 我们晚点再讨论嵌套路由。
const routes = [
  { path: '/foo', component: Foo },
  { path: '/bar', component: Bar }
]

// 3. 创建 router 实例，然后传 `routes` 配置
// 你还可以传别的配置参数，不过先这么简单着吧。
const router = new VueRouter({

```

```
routes // (缩写) 相当于 routes: routes  
})
```

// 4. 创建和挂载根实例。

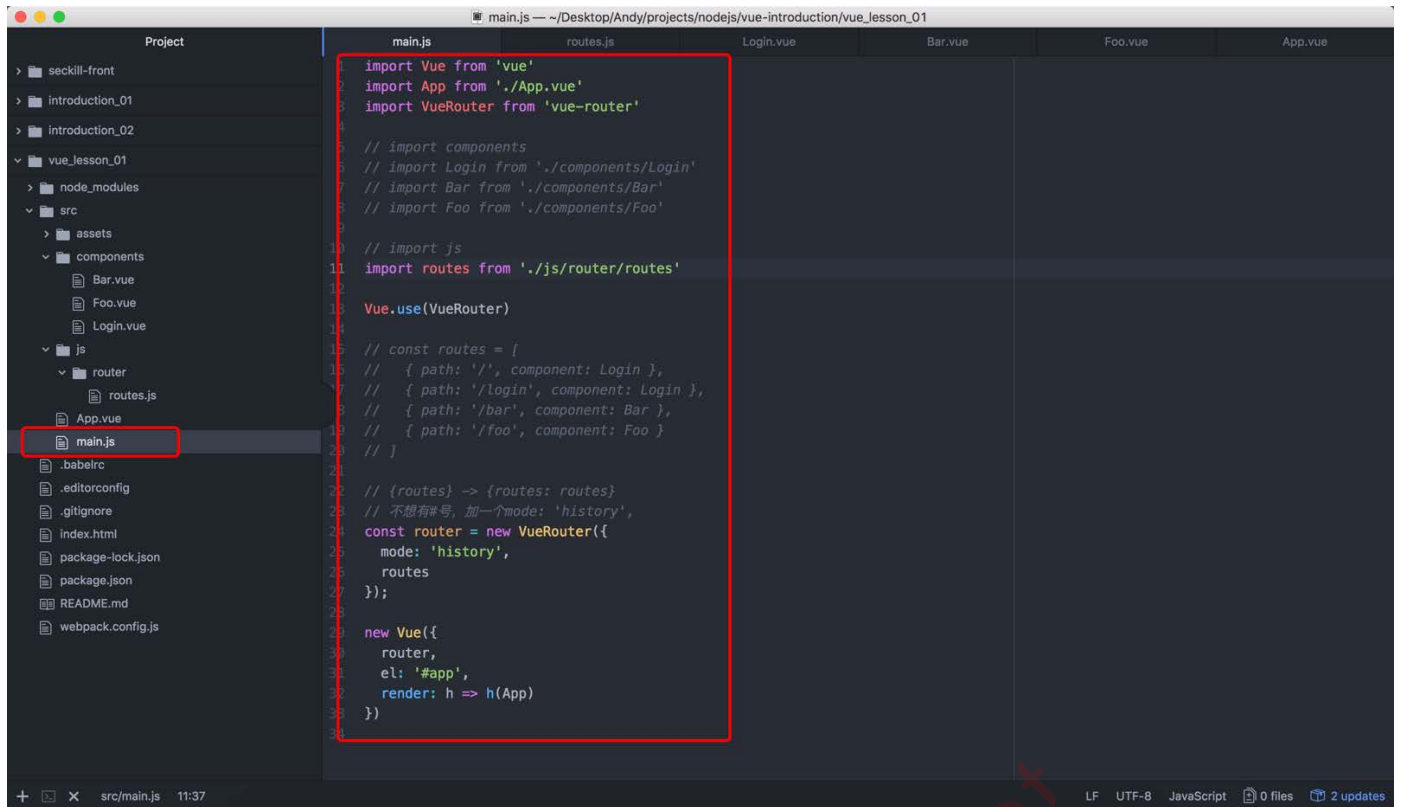
// 记得要通过 router 配置参数注入路由,

// 从而让整个应用都有路由功能

```
const app = new Vue({  
  router  
}).$mount('#app')
```

#### 4. 使用 Vue-router 实现登录页面跳转

```
AndyZhengdeMacBook-Pro:vue_lesson_01 andyzheng$ npm install vue-router -d  
npm info it worked if it ends with ok  
npm info using npm@5.8.0  
npm info using node@v8.11.1  
npm http fetch GET 304 https://registry.npm.taobao.org/vue-router 302ms (from cache)  
npm info lifecycle vue-router@3.0.1~preinstall: vue-router@3.0.1  
npm info linkStuff vue-router@3.0.1  
npm info lifecycle vue-router@3.0.1~install: vue-router@3.0.1  
npm info lifecycle vue-router@3.0.1~postinstall: vue-router@3.0.1  
npm info lifecycle undefined~preshrinkwrap: undefined  
npm info lifecycle undefined~shrinkwrap: undefined  
npm info lifecycle undefined~postshrinkwrap: undefined  
+ vue-router@3.0.1  
added 1 package from 1 contributor in 5.665s  
npm info ok  
AndyZhengdeMacBook-Pro:vue_lesson_01 andyzheng$
```



Project — ~/Desktop/Andy/projects/nodejs/seckill-front-courses/seckill-front

Project

- seckill-front
  - introduction\_01
  - introduction\_02
  - vue\_lesson\_01
    - node\_modules
    - src
      - assets
      - components
        - Bar.vue
        - Foo.vue
        - Login.vue
      - js
        - router
          - routes.js

App.vue

```
1 <template>
2   <div id="app">
3     <!-- 使用 router-link 组件来导航, -->
4     <!-- 通过传入 'to' 属性指定链接, -->
5     <!-- <router-link> 默认会被渲染成一个 'a' 标签 -->
6     <!-- <router-link to="/foo">Go to Foo</router-link> -->
7     <router-link to="/bar">Go to Bar</router-link> -->
8
9     <!-- 需要一些其他业务时 -->
10    <!-- <button @click="gotoByUrl('/foo')">Go to Foo</button> -->
11    <button @click="gotoByUrl('/bar')">Go to Bar</button> -->
12
13    <router-view></router-view>
14  </div>
15 </template>
16
17 <script>
18 export default {
19   name: 'app',
20   data () {
21     return {
22     }
23   },
24   methods: {
25     gotoByUrl(url) {
26       // do some things
27       this.$router.push(url)
28     }
29   }
30 }
31 </script>
32
33 <style lang="scss">
34 </style>
35
```

src/App.vue 27:28 (1, 3)

LF UTF-8 Vue Component 0 files 2 updates

Project — ~/Desktop/Andy/projects/nodejs/seckill-front-courses/seckill-front

Project

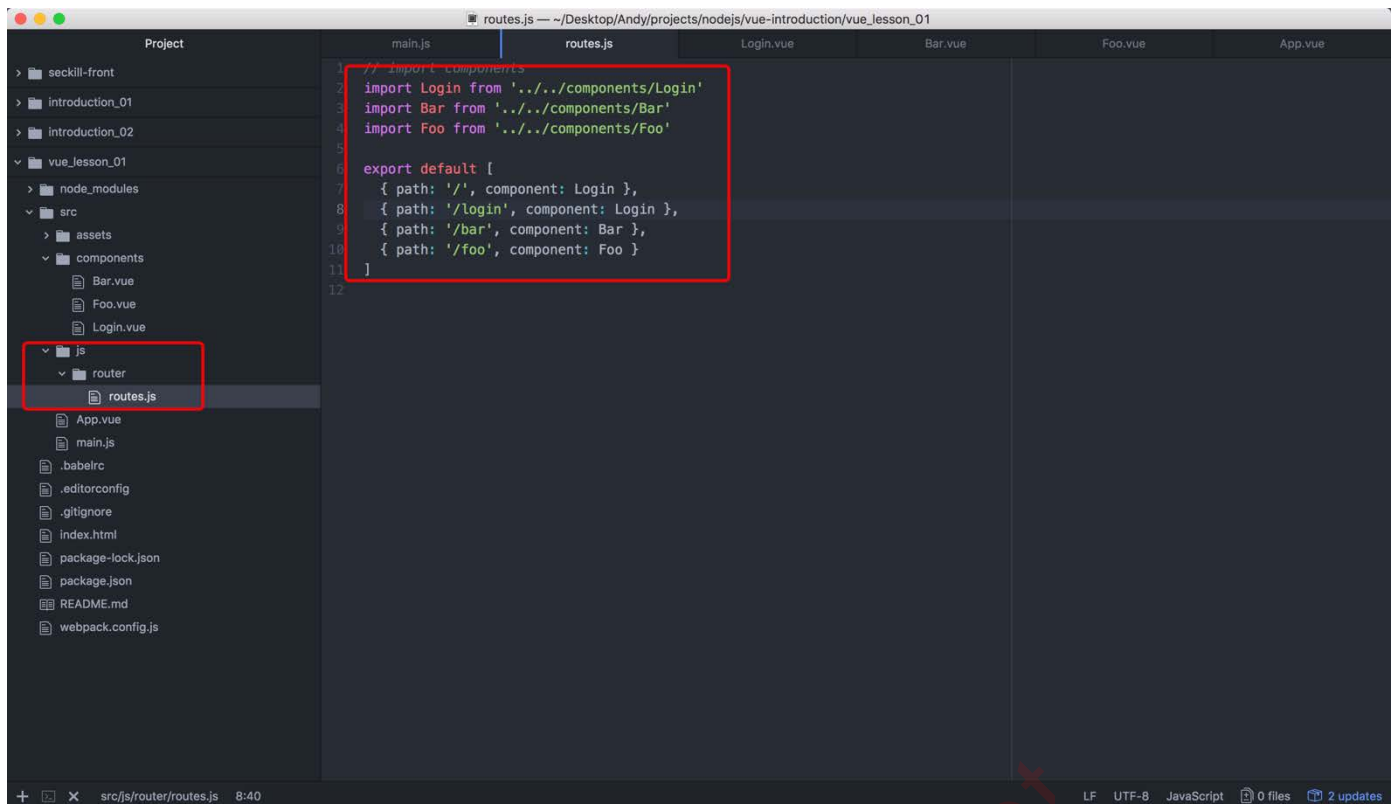
- seckill-front
  - introduction\_01
  - introduction\_02
  - vue\_lesson\_01
    - node\_modules
    - src
      - assets
      - components
        - Bar.vue
        - Foo.vue
        - Login.vue
      - js
        - router
          - routes.js

Login.vue

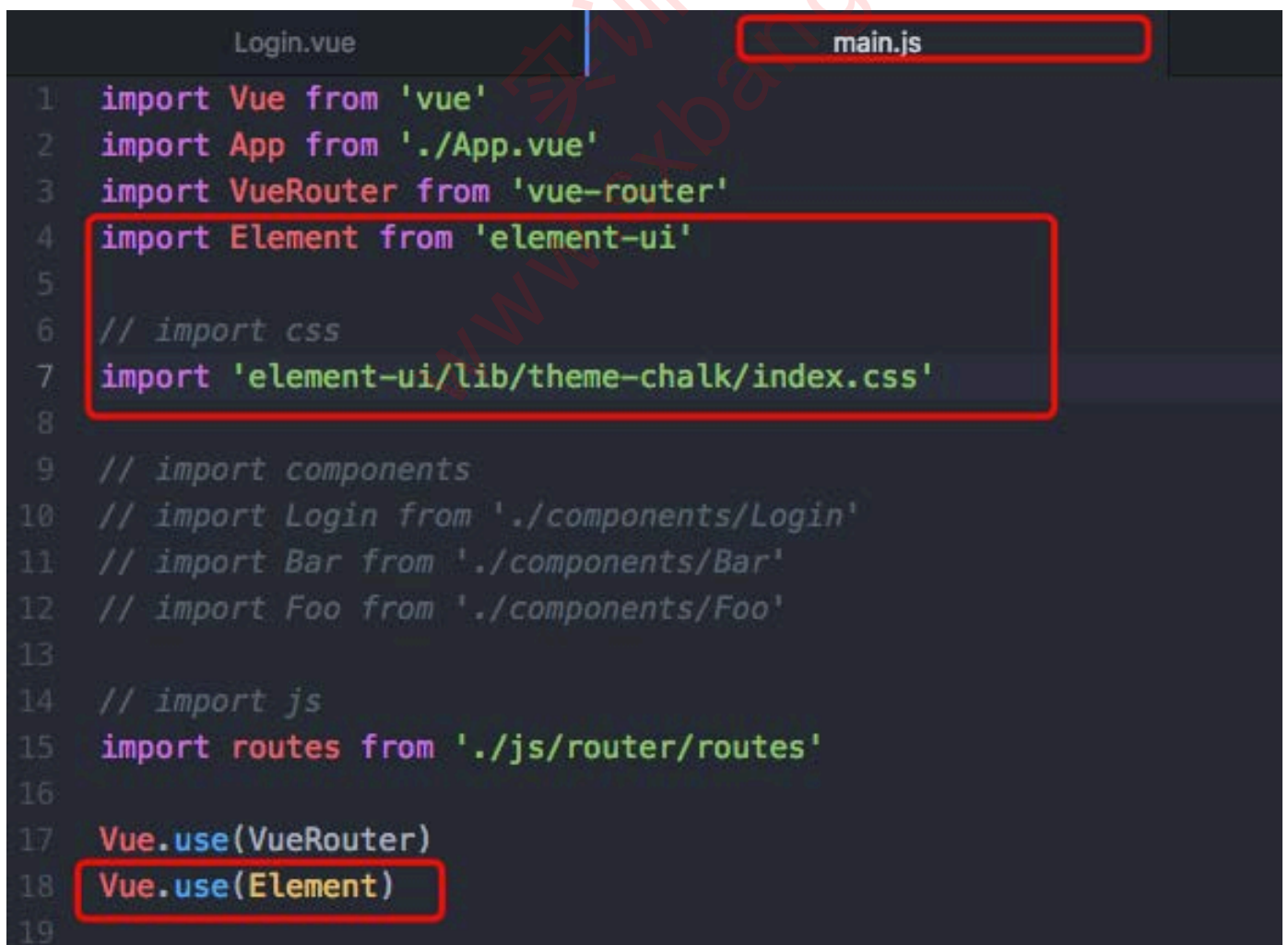
```
1 <template>
2   <div id="login">
3     login
4   </div>
5 </template>
6
7 <script>
8 export default {
9   name: 'login',
10  data () {
11    return {
12    }
13  }
14 }
15 </script>
16
17 <style lang="scss">
18 </style>
19
```

src/components/Login.vue 3:10

LF UTF-8 Vue Component 0 files 2 updates



## 5. 使用 element-ui 完成登录页面布局



Failed to compile.

./node\_modules/element-ui/lib/theme-chalk/fonts/element-icons.ttf

Module parse failed: Unexpected character '' (1:0)

You may need an appropriate loader to handle this file type.

(Source code omitted for this binary file)

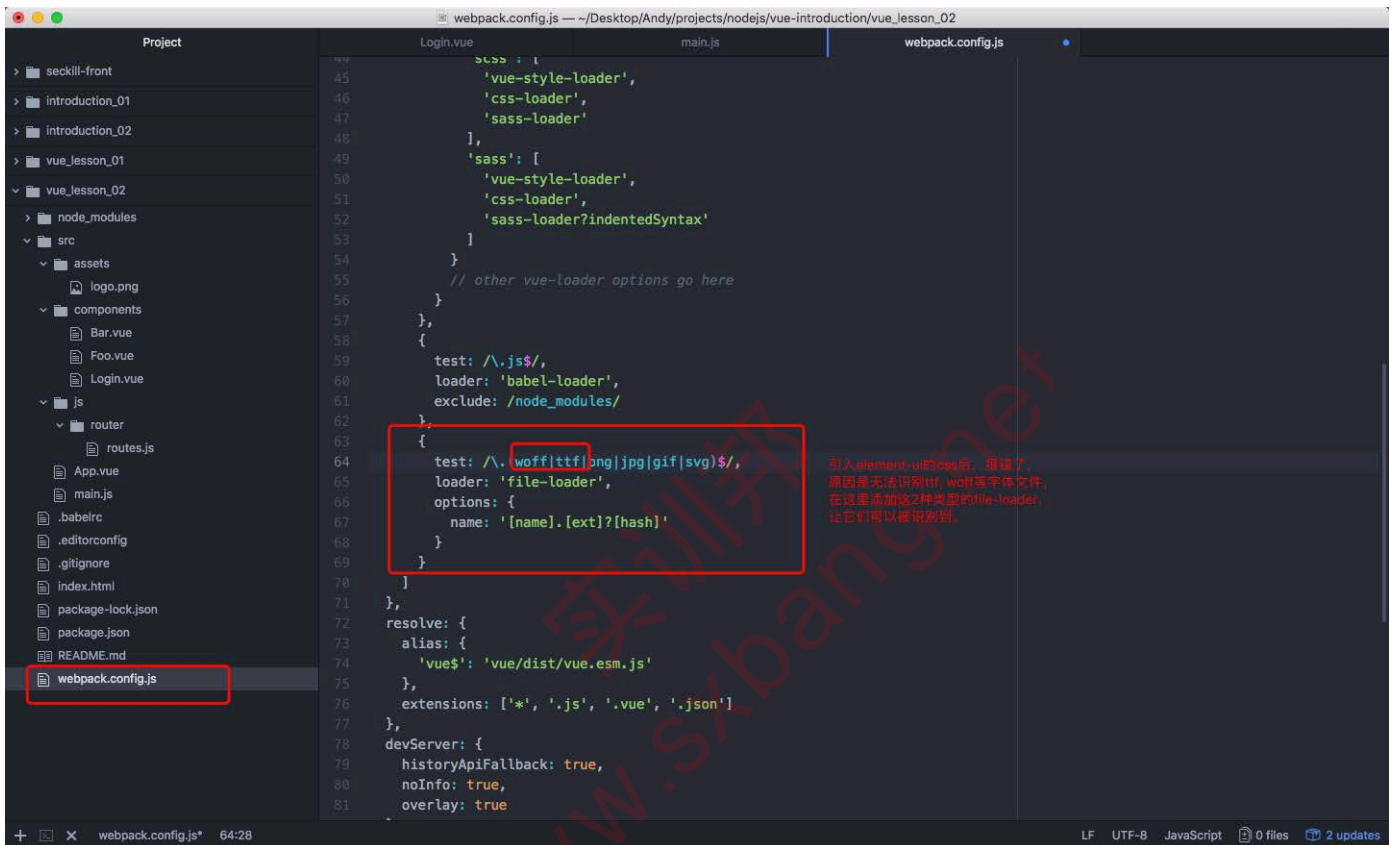
@ ./node\_modules/css-loader!./node\_modules/element-ui/lib/theme-chalk/index.css 7:411-447

@ ./node\_modules/element-ui/lib/theme-chalk/index.css

@ ./src/main.js

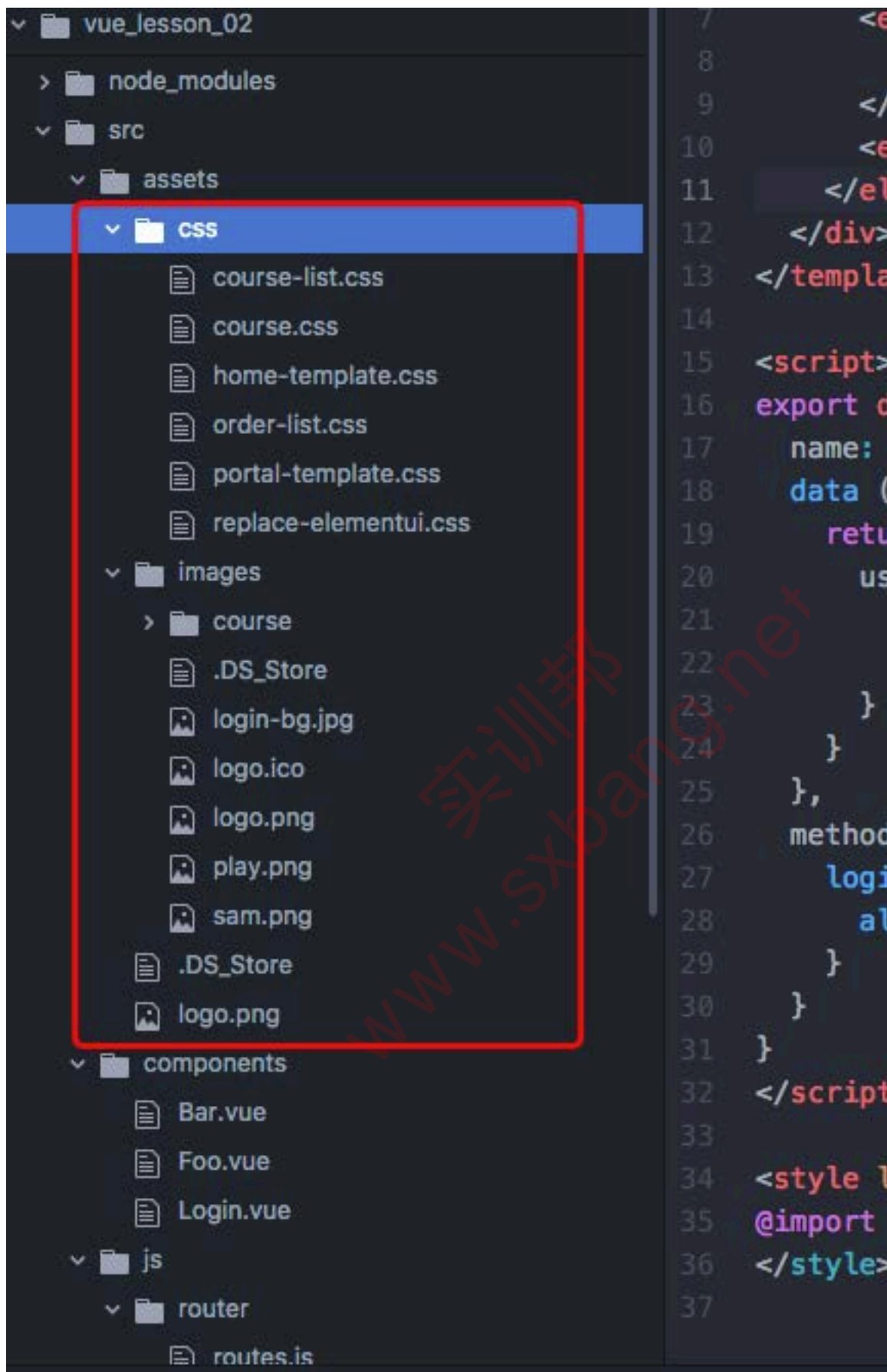
@ multi (webpack)-dev-server/client?http://localhost:8080 webpack/hot/dev-server ./src/main.js

出现以上错误时，说明下图配置未做



做一个简单的登录表单







```
Login.vue App.vue main.js index.html PortalTemplate.vue webpack.config.js
1 <template>
2   <div id="login">
3     <el-form label-position="top" label-width="80px" :model="user">
4       <el-form-item label="用户名">
5         <el-input v-model="user.username" placeholder="请输入用户名" >/el-input>
6       </el-form-item>
7       <el-form-item label="密码">
8         <el-input type="password" v-model="user.password" placeholder="请输入密码" @keypress.enter.native="login" >/el-input>
9       </el-form-item>
10      <el-button type="primary" @click="login">登录</el-button>
11    </el-form>
12  </div>
13
14 </template>
15
16 <script>
17 export default {
18   name: 'login',
19   data () {
20     return {
21       user: {
22         username: '',
23         password: ''
24       }
25     }
26   },
27   methods: {
28     login() {
29       alert(this.user.username + ':' + this.user.password);
30     }
31   }
32 }
33 </script>
34
35 <style lang="scss">
36 @import '../assets/css/portal-template.css'
37 </style>
38
```

完成登录功能UI布局

```
Login.vue App.vue main.js index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>vue_lesson_01</title>
  </head>
  <body class="page-bg">
    <div id="app"></div>
    <script src="/dist/build.js"></script>
  </body>
</html>
```

```
Login.vue | App.vue | main.js | index.html | PortalTemplate.vue | webpack.config.js
</template>

<script>
export default {
  name: 'app',
  data () {
    return {
    }
  }
}
</script>

<style lang="scss">
@import './assets/css/replace-elementui.css';
html {
  height: 100%;
  margin: 0;
  padding: 0;
}
body {
  height: 100%;
  margin: 0;
  padding: 0;
  font-family : "PingFang SC", "Helvetica Neue", "Hiragino Sans GB", "Segoe UI", "Microsoft YaHei", "微软雅黑", sans-serif;
}
#app {
  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
a {
  text-decoration: none;
}
</style>
```

```
@import './assets/css/replace-elementui.css';

html {
  height: 100%;
  margin: 0;
  padding: 0;
}

body {
  height: 100%;
  margin: 0;
  padding: 0;
  font-family : "PingFang SC", "Helvetica Neue", "Hiragino Sans GB", "Segoe UI", "Microsoft YaHei", "微软雅黑", sans-serif;
}

#app {
  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
```

```

justify-content: center;

align-items: center;
}
a {
  text-decoration: none;
}

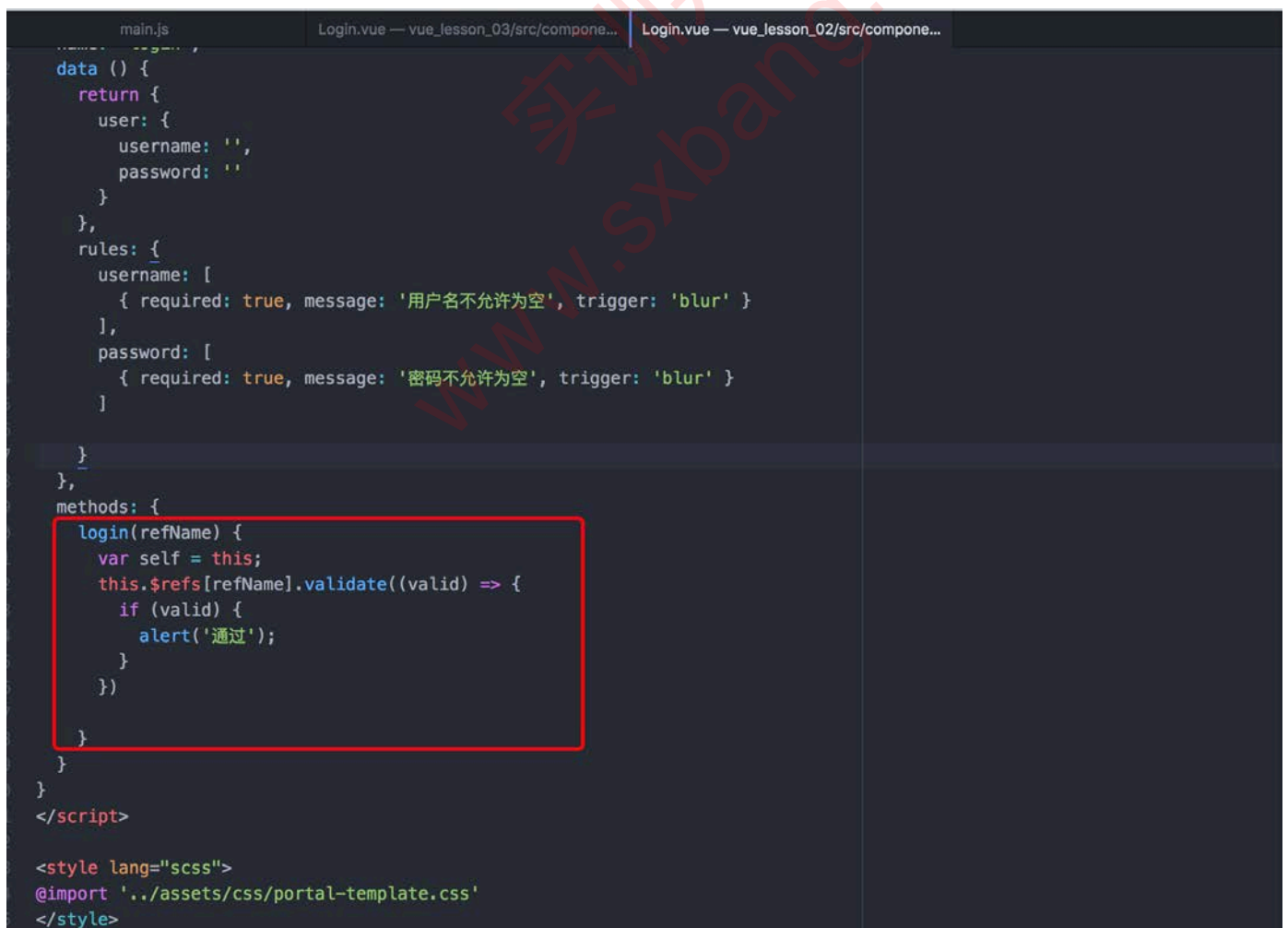
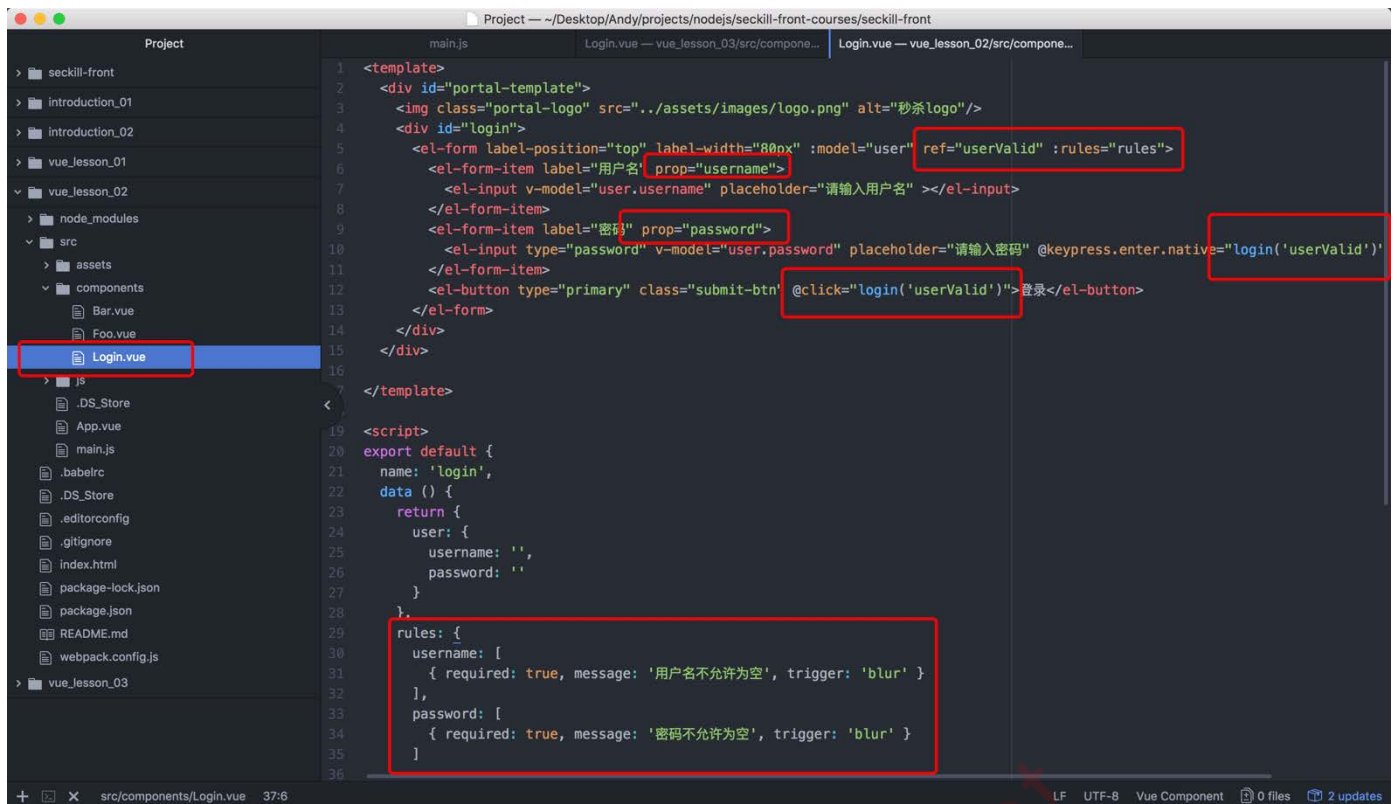
```

```

Login.vue App.vue main.js index.html PortalTemplate.vue webpack.config.js
1 <template>
2   <div id="portal-template">
3     
4     <div id="login">
5       <el-form label-position="top" label-width="80px" :model="user">
6         <el-form-item label="用户名">
7           <el-input v-model="user.username" placeholder="请输入用户名" >/el-input>
8         </el-form-item>
9         <el-form-item label="密码">
10          <el-input type="password" v-model="user.password" placeholder="请输入密码" @keypress.enter.native="login" >/el-input>
11        </el-form-item>
12        <el-button type="primary" class="submit-btn" @click="login">登录</el-button>
13      </el-form>
14    </div>
15  </div>
16 </template>
17
18 <script>
19 export default {
20   name: 'login',
21   data () {
22     return {
23       user: {
24         username: '',
25         password: ''
26       }
27     }
28   },
29   methods: {
30     login() {
31       alert(this.user.username + ':' + this.user.password);
32     }
33   }
34 }
35 </script>
36

```

添加表单验证





THE Milk COURSE

\* 用户名

请输入用户名

用户名不允许为空

\* 密码

请输入密码

密码不允许为空

登录



localhost:8081 显示

通过

确定



THE Milk COURSE

\* 用户名

a

\* 密码

\*

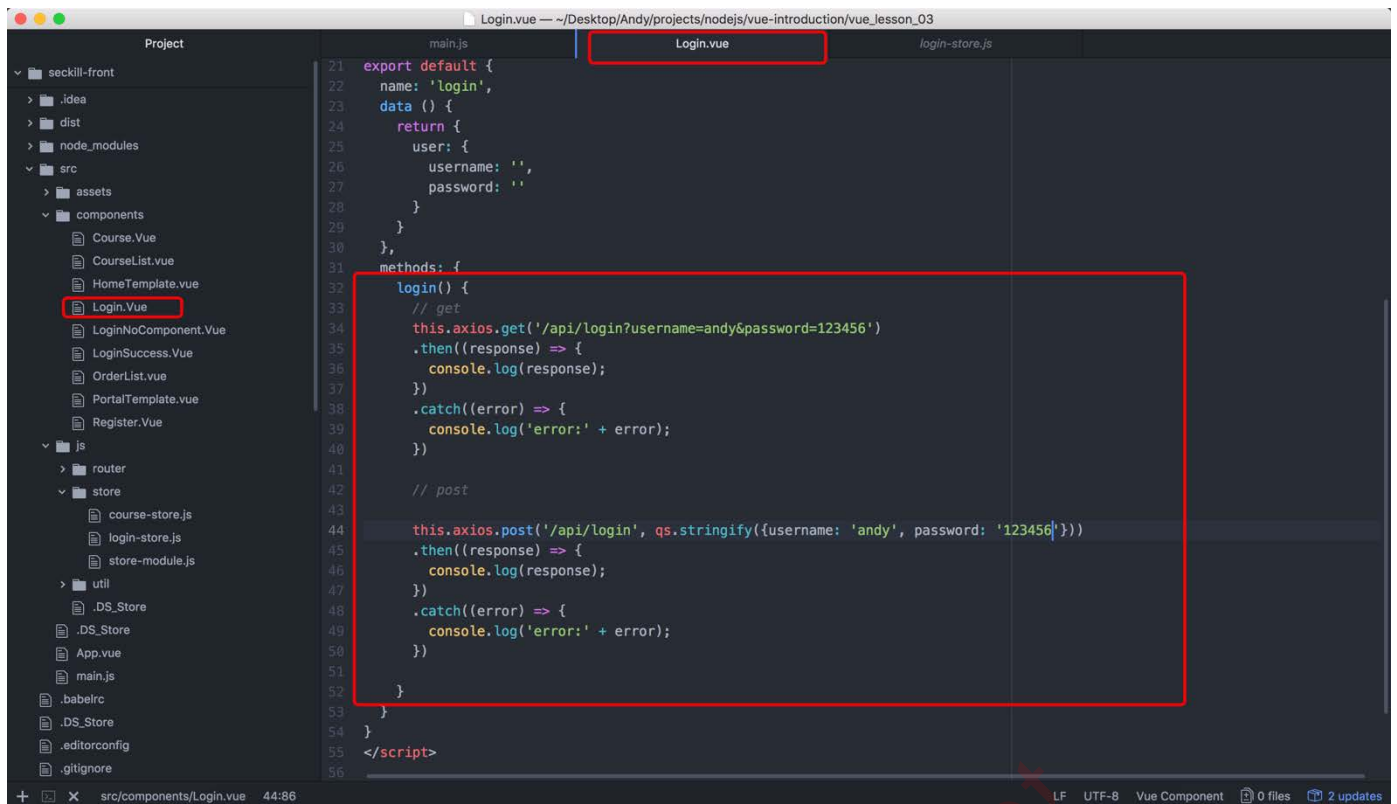
登录

6. 使用 Axios 向后端服务器发送请求进行登录

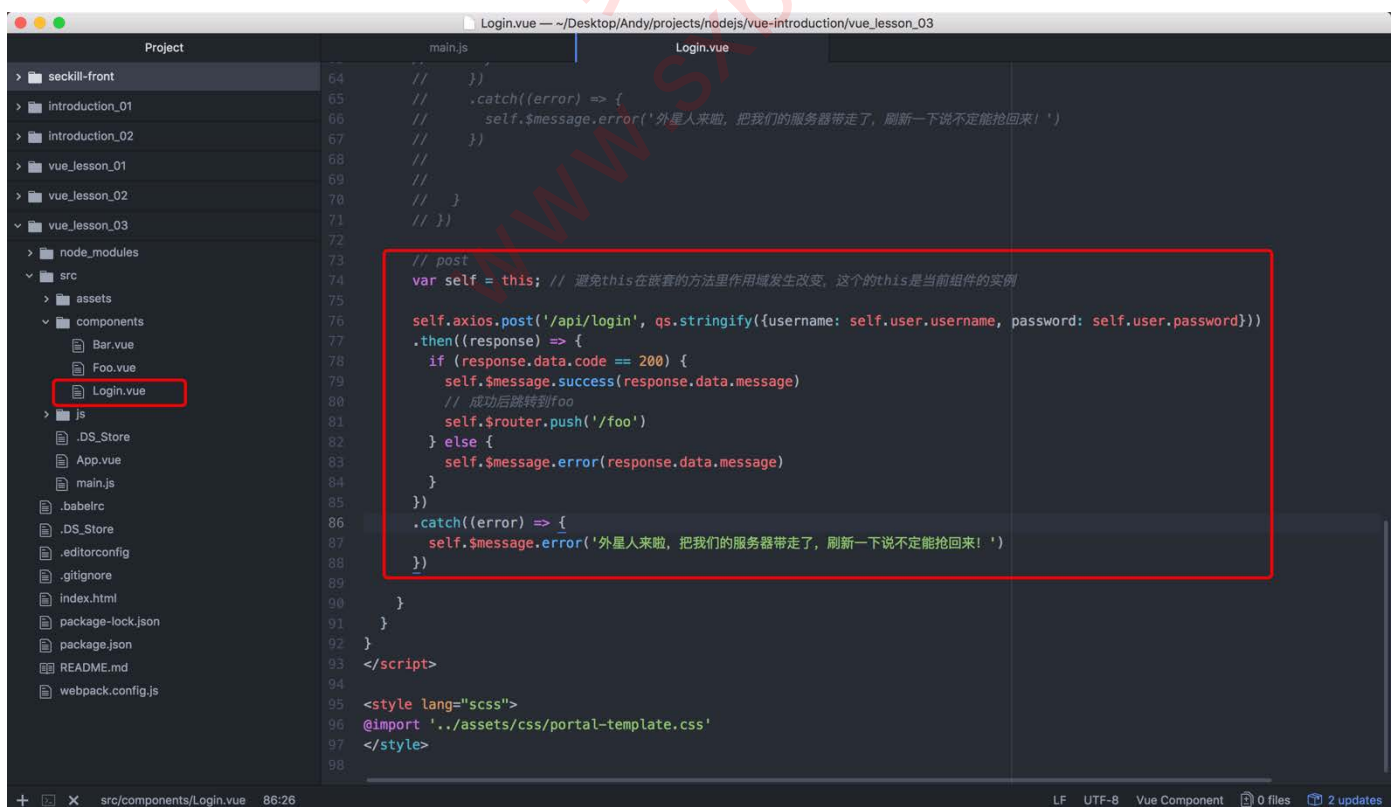
```
73 alias: {
74   'vue$': 'vue/dist/vue.esm.js'
75 },
76 extensions: ['*', '.js', '.vue', '.json']
77 },
78 devServer: {
79   port: '8081',
80   historyApiFallback: true,
81   noInfo: true,
82   overlay: true
83 },
84 performance: {
85   hints: false
86 },
87 devtool: '#eval-source-map'
88 }
89
90 if (process.env.NODE_ENV === 'production') {
91   module.exports.devtool = '#source-map'
92   // http://vue-loader.vuejs.org/en/workflow/production.html
93   module.exports.plugins = (module.exports.plugins || []).concat([
94     new webpack.DefinePlugin({
95       'process.env': {
96         NODE_ENV: '"production"'
```

```
1 import Vue from 'vue'
2 import App from './App.vue'
3 import VueRouter from 'vue-router'
4 import Element from 'element-ui'
5 import Axios from 'axios'
6 import VueAxios from 'vue-axios'
7 import Qs from 'qs'
8
9 // import css
10 import 'element-ui/lib/theme-chalk/index.css'
11 // import js
12 import routes from './js/router/routes'
13
14 Vue.use(VueRouter)
15 Vue.use(Element)
16 Vue.use(Axios, Axios)
17 Vue.use(Qs)
18
19 Axios.defaults.baseURL = 'http://localhost:8080'
20 Axios.defaults.headers['Content-Type'] = 'application/x-www-form-urlencoded;charset=UTF-8'
21 Axios.defaults.withCredentials = true
22
23 // {routes} -> {routes: routes}
24 // 不想有#号, 加一个mode: 'history',
25 const router = new VueRouter({
26   mode: 'history',
27   routes
28 });
29
30 new Vue({
31   router,
32   el: '#app',
33   render: h => h(App)
34 })
35
```





完成登录功能API请求处理



## 添加表单验证

```
main.js      Login.vue

3  
4  <div id="login">
5    <el-form label-position="top" label-width="80px" :model="user" ref="userValid" :rules="rules">
6      <el-form-item label="用户名" prop="username">
7        <el-input v-model="user.username" placeholder="请输入用户名" ></el-input>
8      </el-form-item>
9      <el-form-item label="密码" prop="password">
10       <el-input type="password" v-model="user.password" placeholder="请输入密码" @keypress.enter.native="login('userValid')">
11     </el-form-item>
12     <el-button type="primary" class="submit-btn" @click="login('userValid')">登录</el-button>
13   </el-form>
14 </div>
15 </div>
16
17 </template>
18
19 <script>
20 import qs from 'qs'
21 export default {
22   name: 'login',
23   data () {
24     return {
25       user: {
26         username: '',
27         password: ''
28       },
29       rules: {
30         username: [
31           { required: true, message: '用户名不允许为空', trigger: 'blur' }
32         ],
33         password: [
34           { required: true, message: '密码不允许为空', trigger: 'blur' }
35         ]
36       }
37     }
38   }
39 }
```

成功

foo



❌ 登录失败，用户名或密码出错，请重新输入。

THE Milk COURSE

\* 用户名

andy

\* 密码

\*\*\*

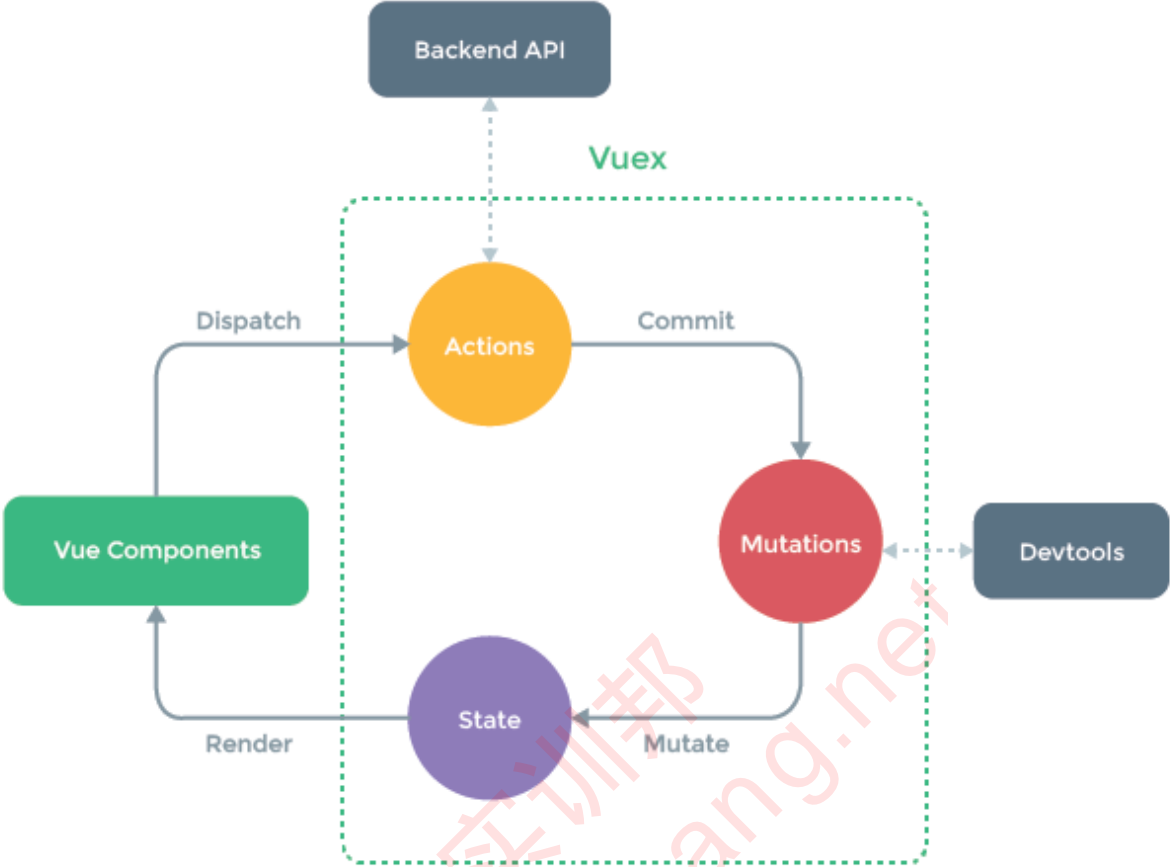
登录

## 7. 使用 Vuex 管理登录状态

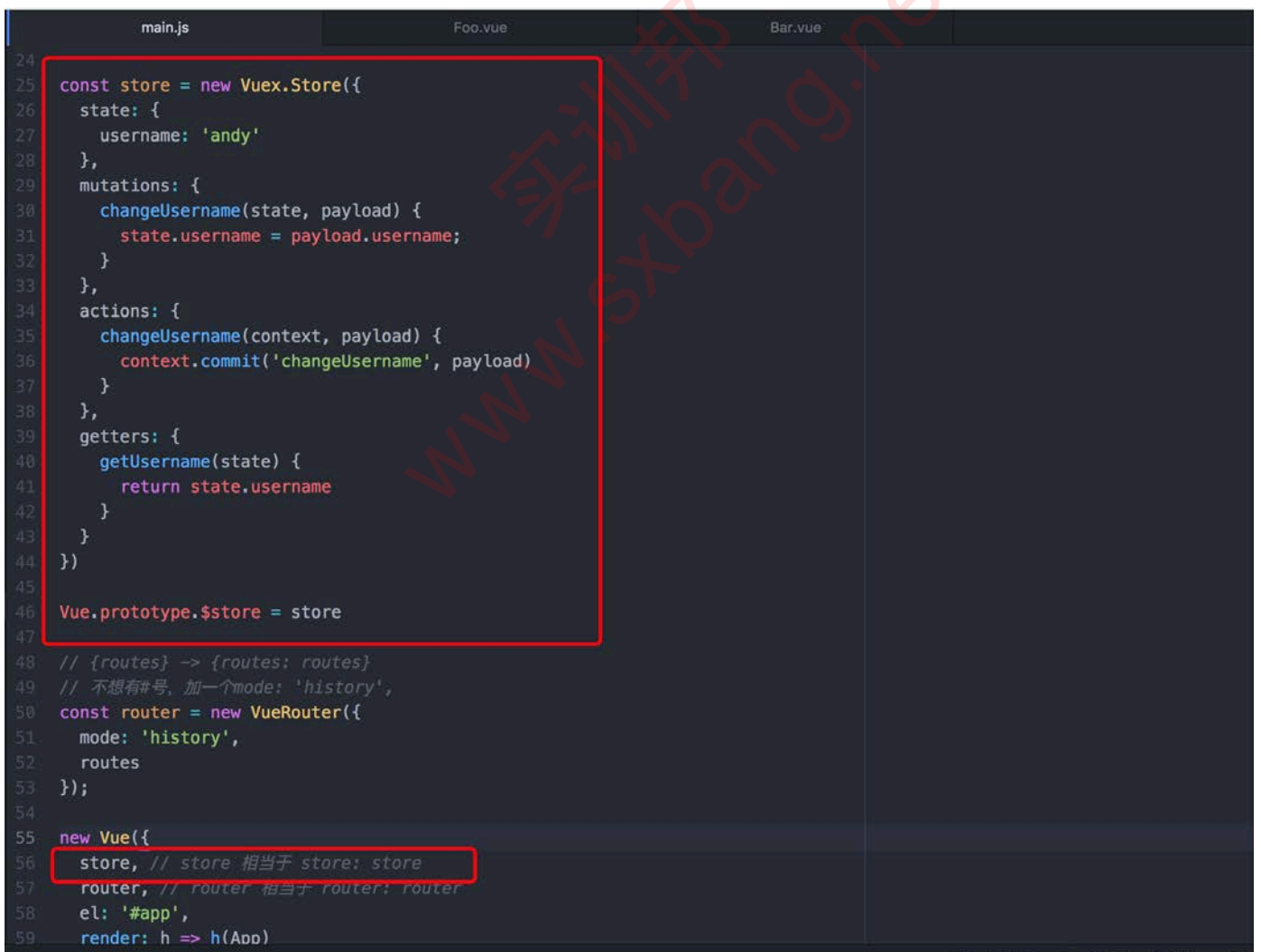
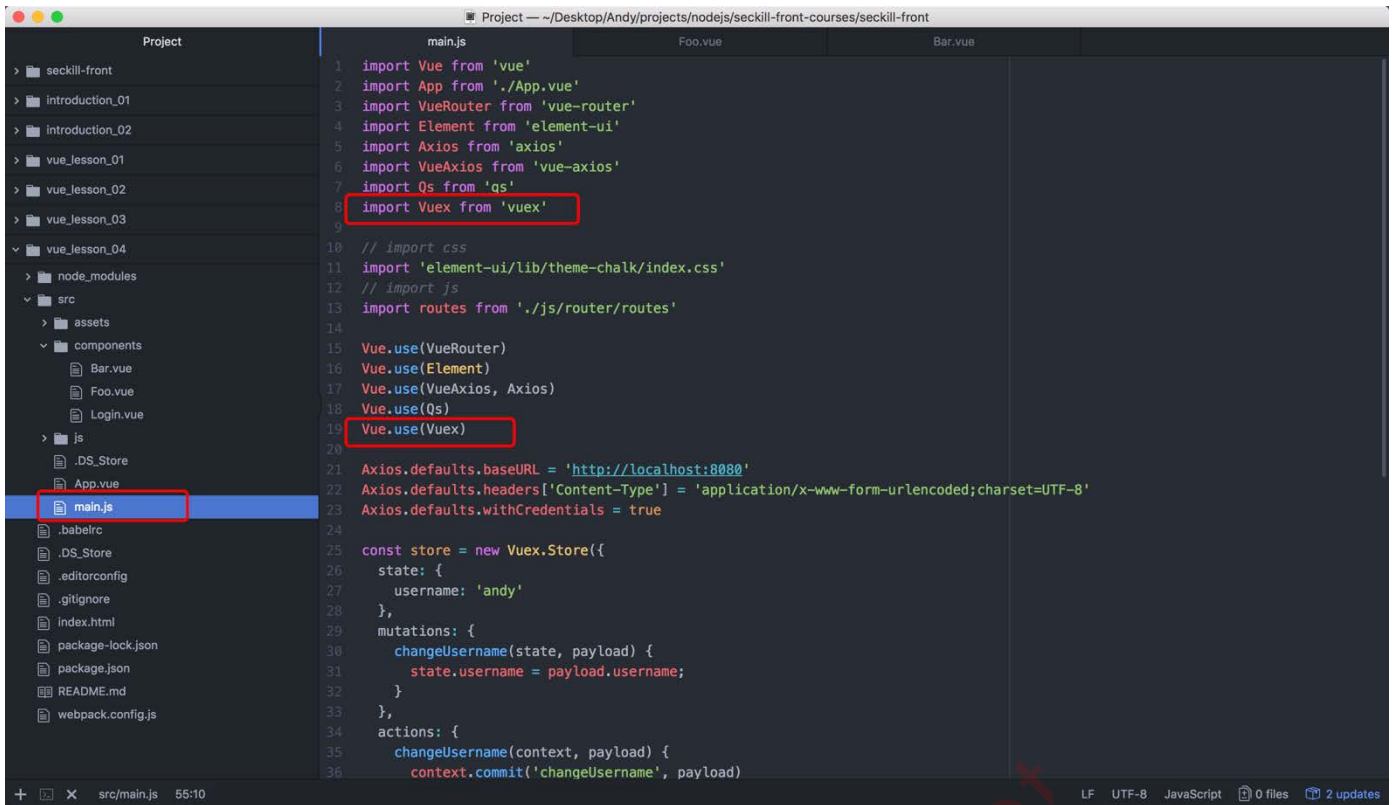
- vuex基本使用

实训邦  
www.sxbang.net

流程



使用



Project

- seckill-front
- introduction\_01
- introduction\_02
- vue\_lesson\_01
- vue\_lesson\_02
- vue\_lesson\_03
- vue\_lesson\_04
  - node\_modules
  - src
    - assets
    - components
      - Bar.vue
      - Foo.vue**
      - Login.vue
    - js
      - .DS\_Store
      - App.vue
      - main.js
    - .babelrc
    - .DS\_Store
    - .editorconfig
    - .gitignore
    - index.html
    - package-lock.json
    - package.json
    - README.md
    - webpack.config.js

main.js    Foo.vue    Bar.vue

```
1 <template>
2   <div id="foo">
3     foo ---- {{getUsername}}
4     <button @click="changeUsername">change username to 'foo'</button>
5   </div>
6 </template>
7
8 <script>
9 export default {
10   name: 'foo',
11   data () {
12     return {
13     },
14   },
15   methods: {
16     changeUsername: function() {
17       this.$store.dispatch('changeUsername', {username: 'foo'})
18       this.$router.push('/bar')
19     }
20   },
21   computed: {
22     getUsername: function() {
23       return this.$store.getters.getUsername
24     }
25   }
26 }
27 </script>
28
29 <style lang="scss">
30 </style>
31
```

src/components/Foo.vue 28:1    LF    UTF-8    Vue Component    0 files    2 updates

Project

- seckill-front
- introduction\_01
- introduction\_02
- vue\_lesson\_01
- vue\_lesson\_02
- vue\_lesson\_03
- vue\_lesson\_04
  - node\_modules
  - src
    - assets
    - components
      - Bar.vue**
      - Foo.vue
      - Login.vue
    - js
      - .DS\_Store
      - App.vue
      - main.js
    - .babelrc
    - .DS\_Store
    - .editorconfig
    - .gitignore
    - index.html
    - package-lock.json
    - package.json
    - README.md
    - webpack.config.js

main.js    Foo.vue    Bar.vue

```
1 <template>
2   <div id="bar">
3     bar ---- {{getUsername}}
4   </div>
5 </template>
6
7 <script>
8 export default {
9   name: 'bar',
10   data () {
11     return {
12     },
13   },
14   computed: {
15     getUsername: function() {
16       return this.$store.getters.getUsername
17     }
18   }
19 }
20 </script>
21
22 <style lang="scss">
23 </style>
24
```

src/components/Bar.vue 20:10    LF    UTF-8    Vue Component    0 files    2 updates

foo ---- andy

bar ---- foo

- 完成登录功能的状态管理

login-store.js — ~/Desktop/Andy/projects/nodejs/vue-introduction/vue\_lesson\_04

Project

- seckill-front
- introduction\_01
- introduction\_02
- vue\_lesson\_01
- vue\_lesson\_02
- vue\_lesson\_03
- vue\_lesson\_04
  - node\_modules
  - src
    - assets
    - components
      - Bar.vue
      - Foo.vue
      - Login.vue
    - js
      - router
        - routes.js
      - store
        - login-store.js
        - store-module.js
      - .DS\_Store
      - App.vue
      - main.js
      - .babelrc
      - .DS\_Store
      - .editorconfig
      - .gitignore
      - index.html
      - package-lock.json

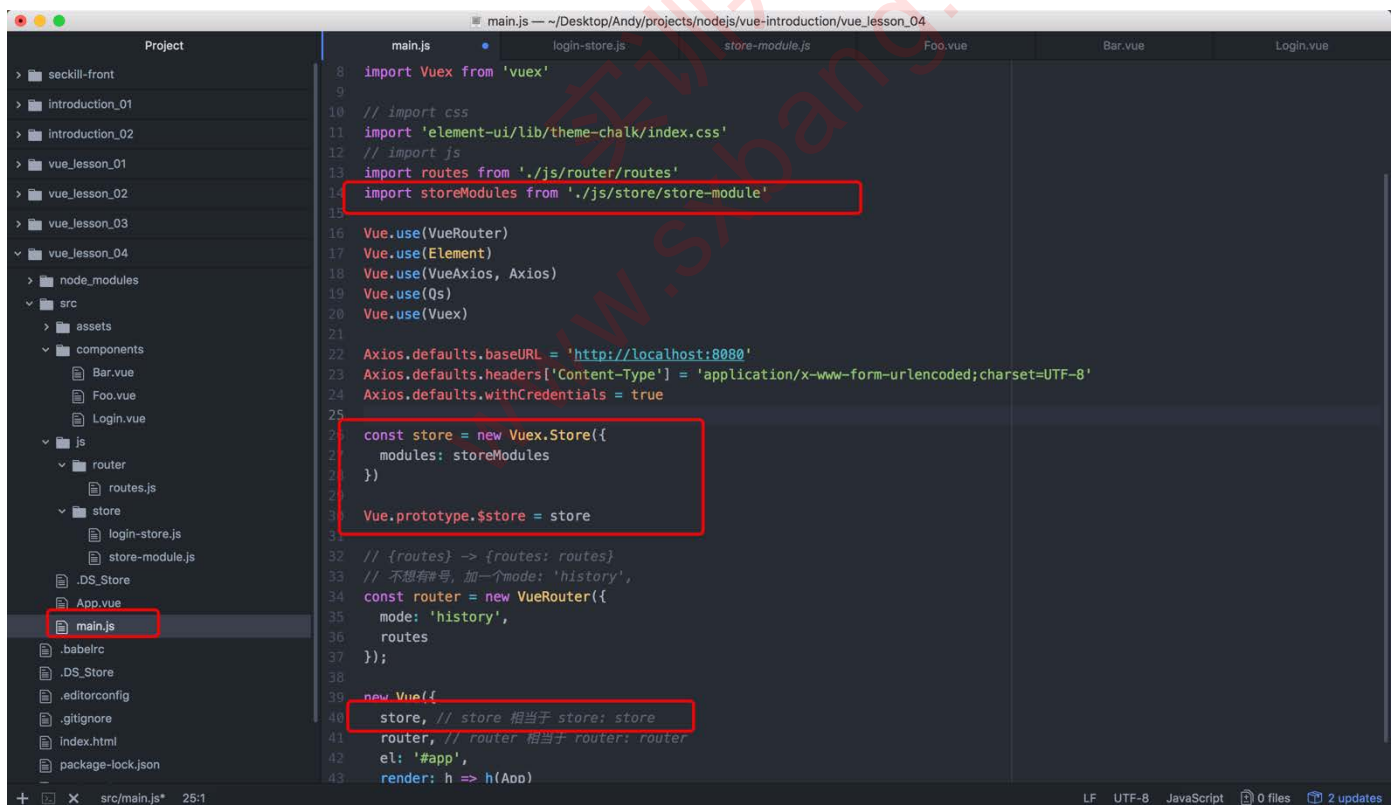
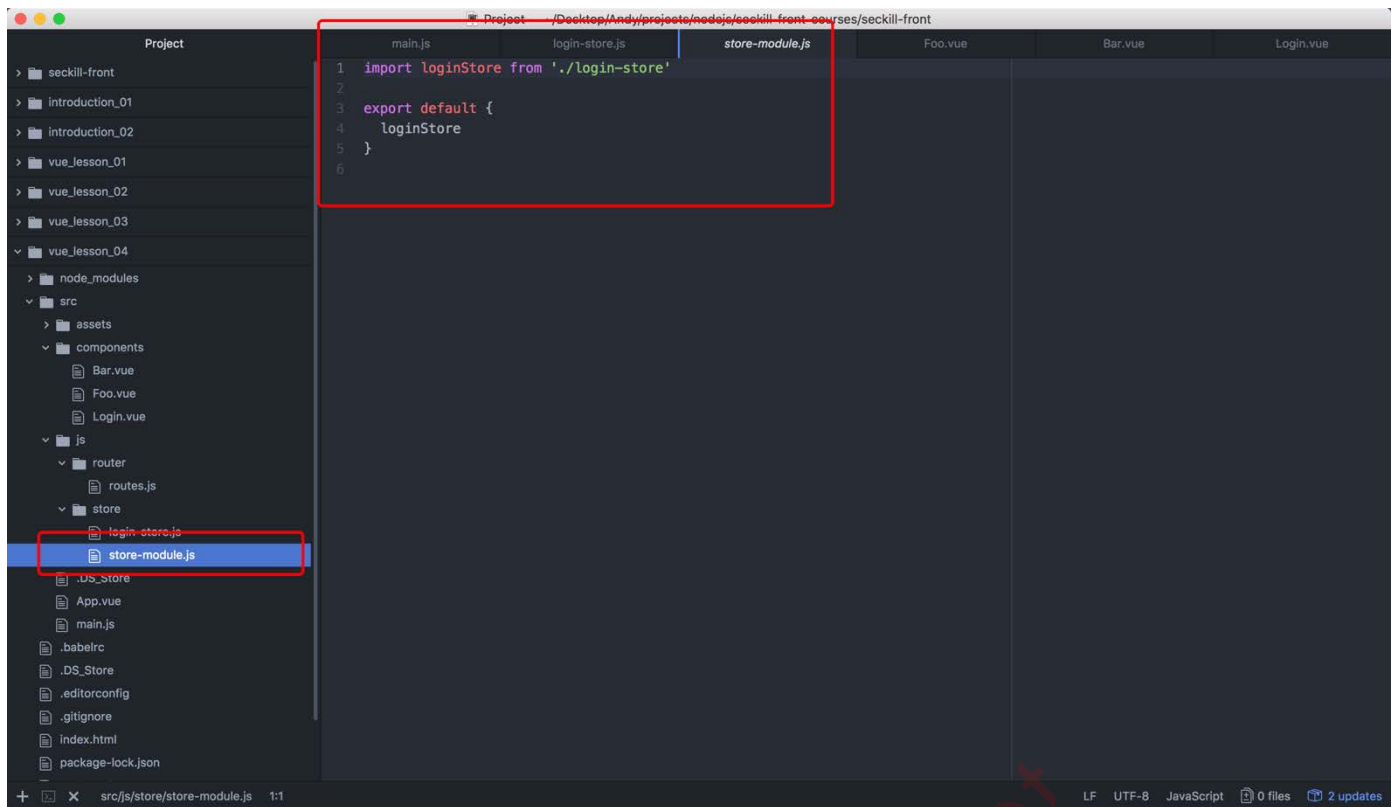
```
1 import Vue from 'vue'
2 import qs from 'qs'
3
4 export default {
5   state: {
6     login: false,
7     user: {
8       username: '',
9       password: ''
10    },
11  },
12  mutations: {
13    login(state, payload) {
14      sessionStorage.setItem('username', payload.username);
15      sessionStorage.setItem('login', true);
16      state.login = true;
17      state.user.username = payload.username;
18    },
19  },
20  actions: {
21    login(context, payload) {
22      return new Promise((resolve, reject) => {
23        Vue.axios.post('/api/login',
24          qs.stringify(payload))
25          .then(function(response) {
26            if (response.data.code == 200) {
27              resolve(response)
28              context.commit('login', payload);
29            } else {
30              reject(response)
31            }
32          })
33          .catch(function(error) {
34            var errorData = {
35              message: '外星人来啦，把我们的服务器带走了，刷新一下说不定能抢回来！'
36            }
37          })
38      })
39    },
40  },
41  getters: {
42    getLoginState(state) {
43      return state.login
44    },
45    getUsername(state) {
46      return state.user.username
47    }
48  }
49 }
```

src/js/store/login-store.js 47:7

LF UTF-8 JavaScript 0 files 2 updates

main.js login-store.js Foo.vue Bar.vue

```
18 }
19 },
20 actions: {
21   login(context, payload) {
22     return new Promise((resolve, reject) => {
23       Vue.axios.post('/api/login',
24         qs.stringify(payload))
25       .then(function(response) {
26         if (response.data.code == 200) {
27           resolve(response)
28           context.commit('login', payload);
29         } else {
30           reject(response)
31         }
32       })
33       .catch(function(error) {
34         var errorData = {
35           message: '外星人来啦，把我们的服务器带走了，刷新一下说不定能抢回来！'
36         }
37         error.data = errorData
38         reject(error)
39       })
40     })
41   },
42 },
43 },
44 getters: {
45   getLoginState(state) {
46     return state.login
47   },
48   getUsername(state) {
49     return state.user.username
50   }
51 }
52 }
```

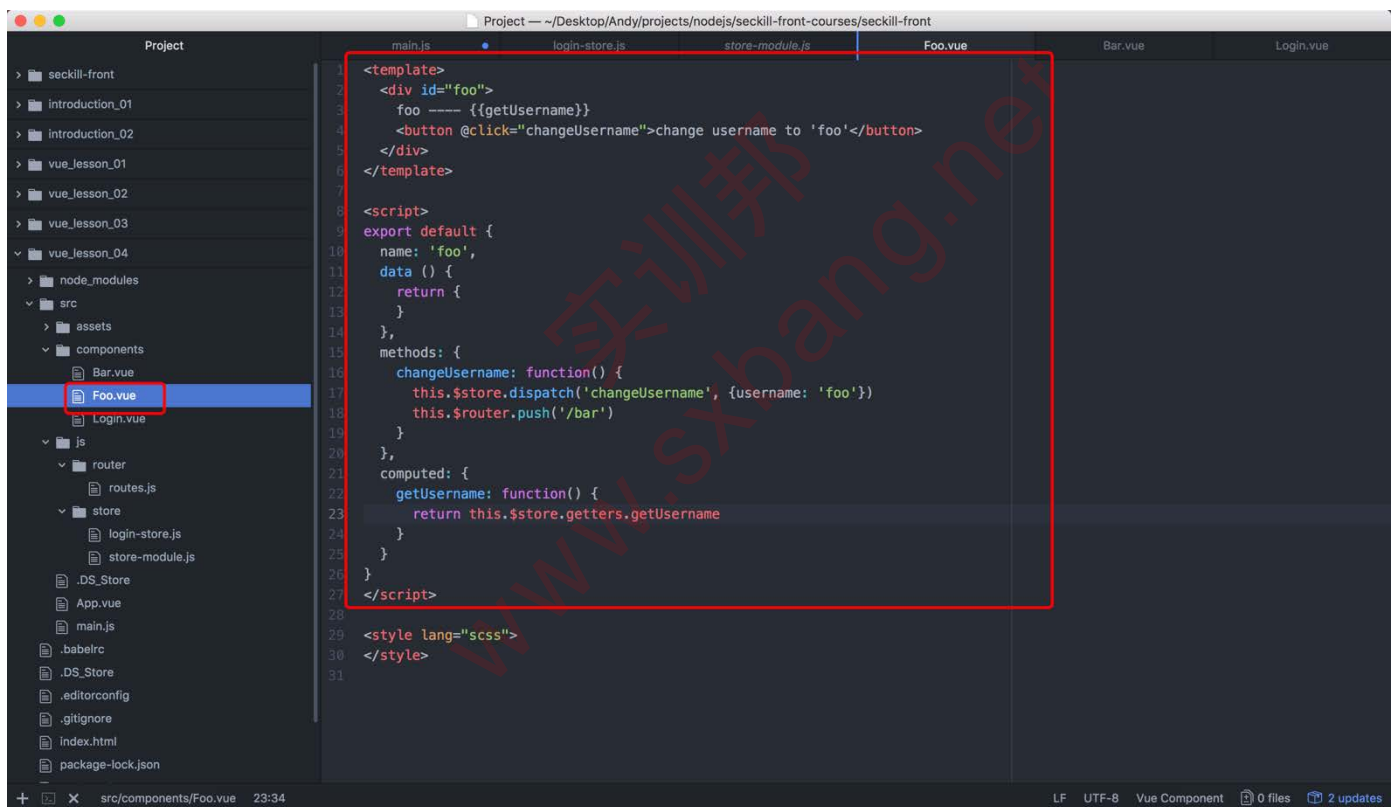




```

login(refName) {
  var self = this;
  this.$refs[refName].validate((valid) => {
    if (valid) {
      this.$store.dispatch('login', {
        username: this.user.username,
        password: this.user.password
      })
      .then((response) => {
        self.$router.push('/foo')
        self.$message.success(response.data.message)
      })
      .catch((error) => {
        self.$message.error(error.data.message)
      })
    }
  });
}

```







foo ----



✖ 登录失败，用户名或密码出错，请重新输入。

THE Milk COURSE

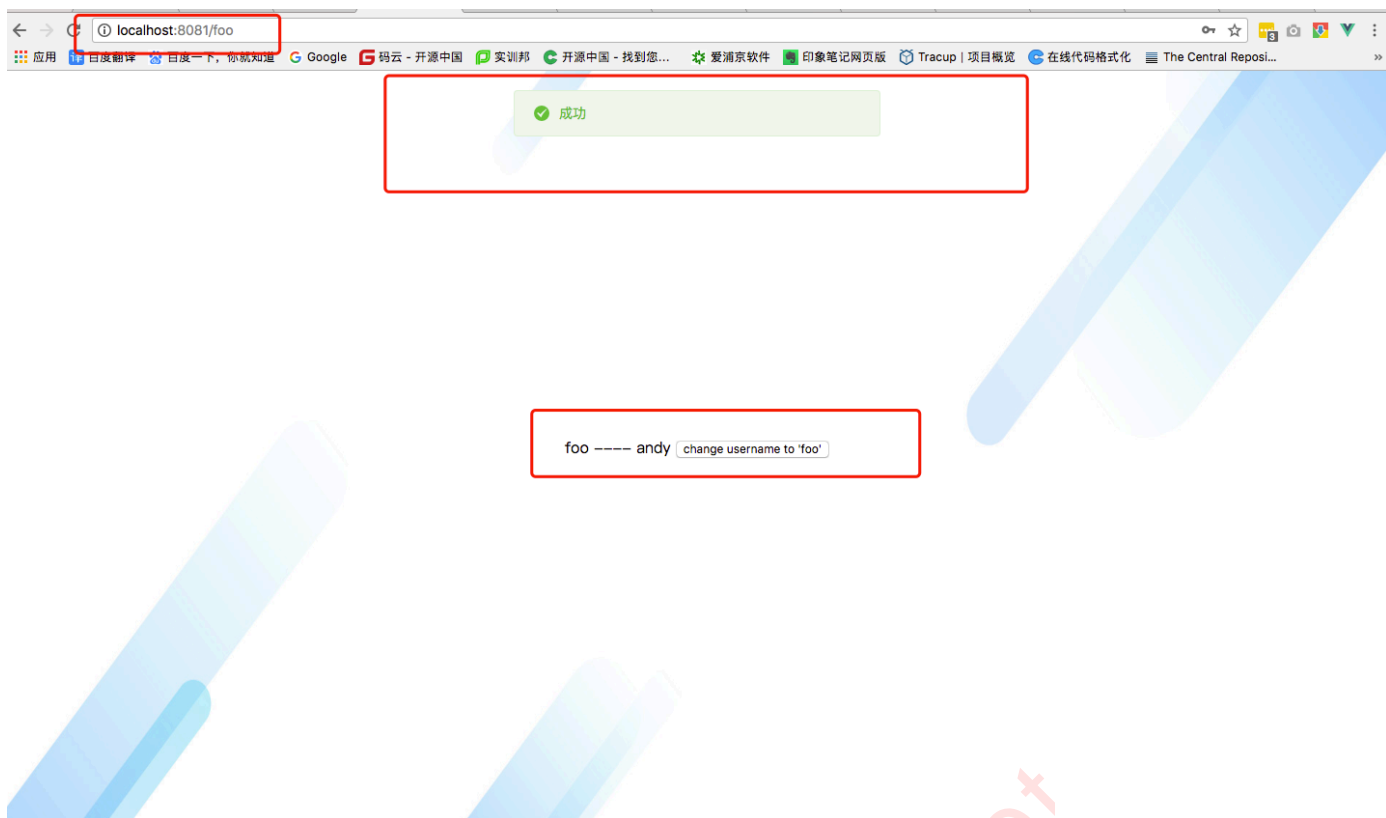
\* 用户名

andy

\* 密码

\*\*\*\*\*

登录



## 源码参考

N/A

## 回马枪总结

1. 导入组件，配置路由；
2. 使用 Element UI 进行布局；
3. 使用 Axios 请求 API 获取后台数据，与输入信息一致则跳转到课程列表。
4. 使用 Vuex 管理登录状态