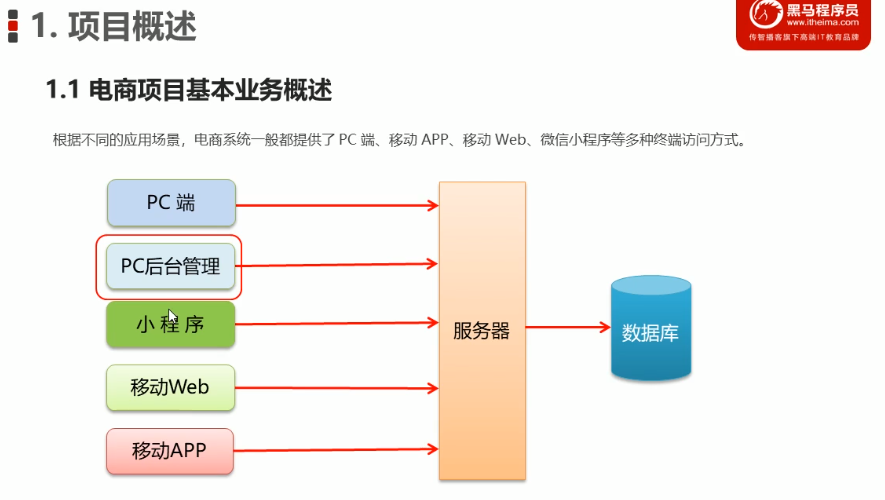
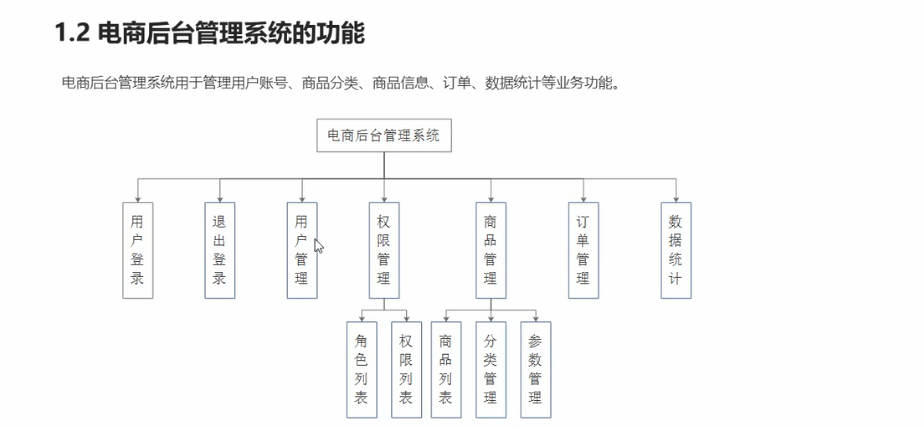
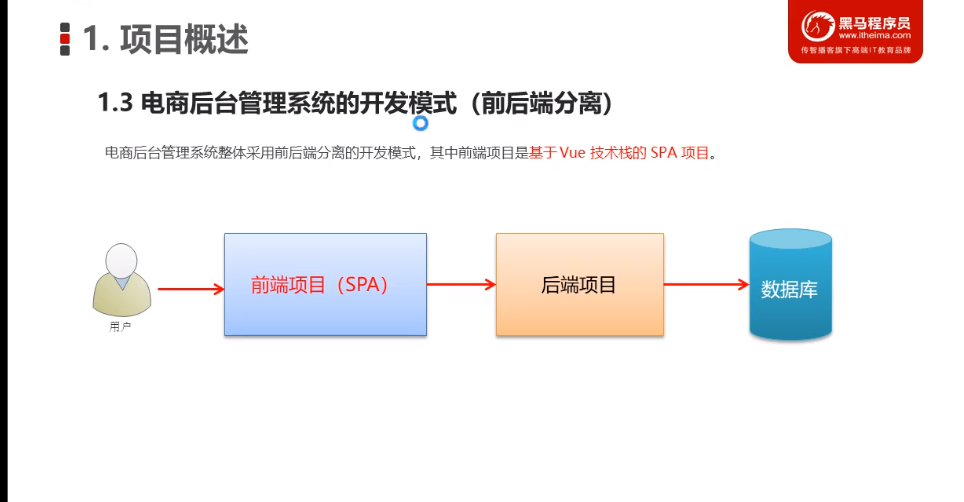
## 学习目标及大纲

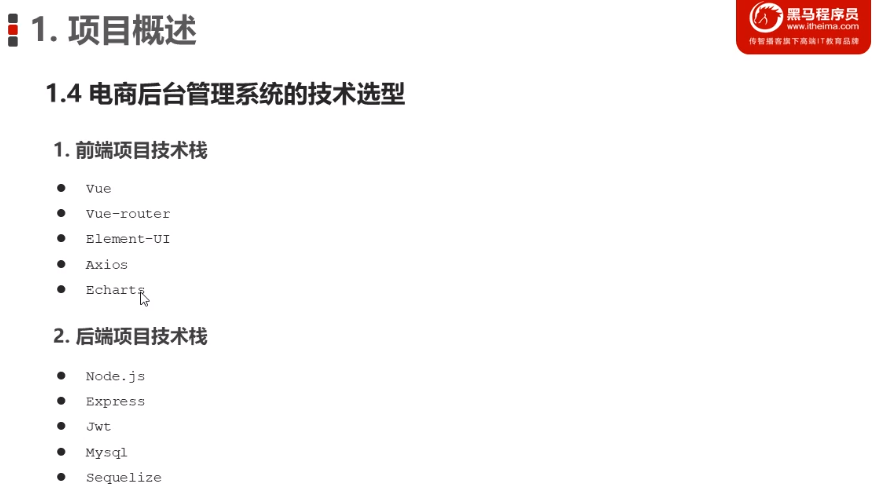


## 项目概述







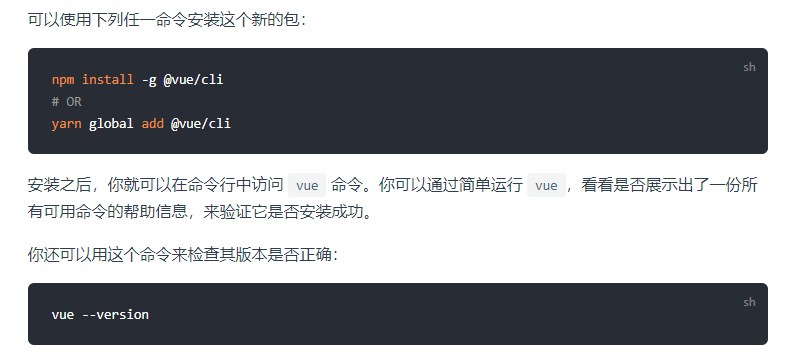


jwt 状态保持，类似于session,sequelize js数据库操作库

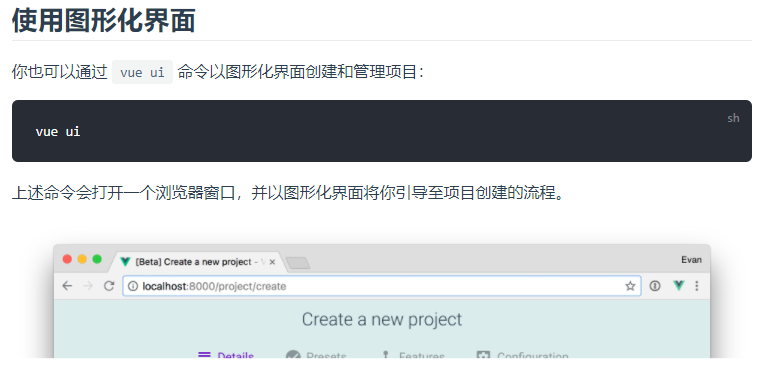
## 项目初始化



### 安装vue手脚架



### 通过vue手脚架创建项目



### 项目的一些基本配置

选择手动配置



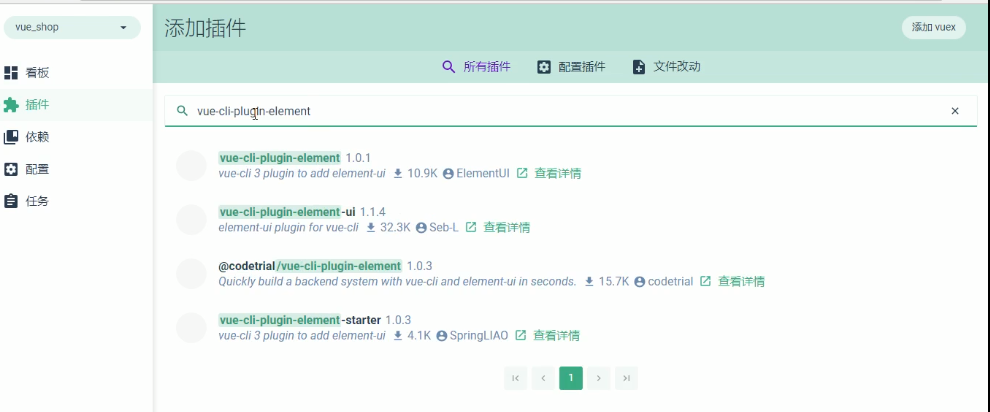
安装必要的功能



必须安装的功能 babel,router,linter/Fomatter(代码格式化工具)，选择使用配置文件，其他配置如下图



### 安装 element-ui插件



### 安装 axio依赖



## 登录功能

### token原理



### 分析页面布局

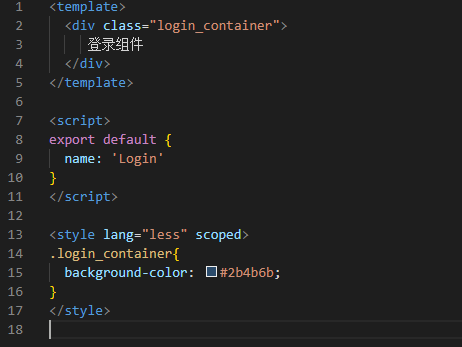
在开发新的功能时，一般新建一个分支，git checkout -b newBranchName

梳理项目的基本结构

将项目中不必要的组件和文件，以及router中的内容，都删掉，保持一个干净的目录结构。

### 设置背景色并在中间添加登录框

1. 设置登录页面的背景颜色

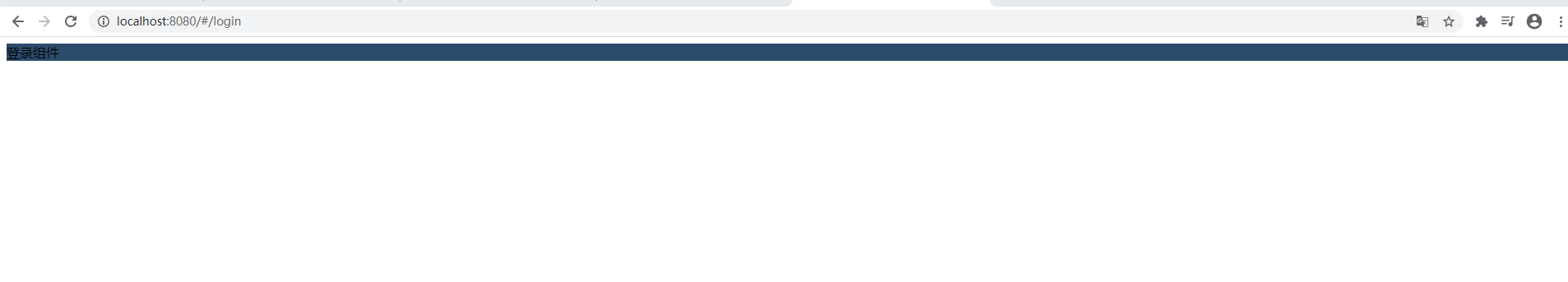


保存后报错，缺少less-loader，vue-cli创建的项目默认不安装less-loader,需要手动安装 less-loader 依赖项。

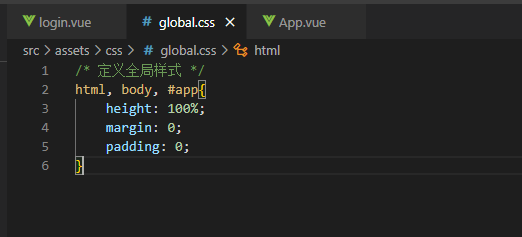
注意less-loader安装到开发依赖中。

Less-loader依赖less,所以后面还要安装less,也是开发依赖。

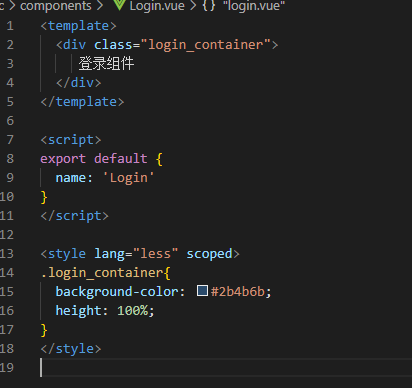
2. 设置页面全屏，定义全局样式



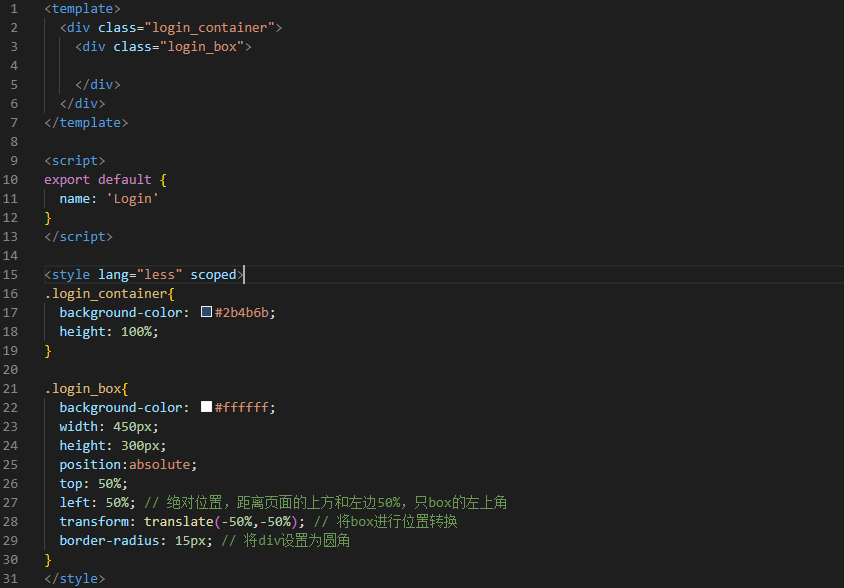
安装好依赖后，发现组件并未全屏展示，打开调试模式查看发现，html元素，body元素及app元素均不为全屏显示。需要设置css样式，在assets文件夹下创建css文件。设置全局样式，并在main.js引入全局样式。



，之后再login\_container的css样式中加入高度100%，整个页面的背景颜色就改变了。



### 设置登录框的样式和位置



### 设置头像的样式和位置

.login\_box{

  background-color: #ffffff;

  width: 450px;

  height: 300px;

  position:absolute;

  top: 50%;

  left: 50%;    // 绝对位置，距离页面的上方和左边50%，只box的左上角

  transform: translate(-50%,-50%);     // 将box进行位置转换

  border-radius: 15px;    // 将div设置为圆角

  box-shadow: 0 0 10px;   //向外扩散10像素的阴影

  .header\_box{    // less嵌套语法,实验放在内外效果一样

    height: 130px;

    width: 130px;

    border: 1px solid #dddddd;    //给header\_box添加边框

    border-radius: 50%;     //将边框设置为圆角

    padding: 10px;    //内间距，图片和外部box的间距

    position: absolute;

    left: 50%;

    transform: translate(-50%,-50%);    //将外面的div位置摆到中间

    background-color: #ffffff;

    img{

      width: 100%;  //这 里的100%是以父盒子 header\_box 为参照

      height: 100%;

      border-radius: 50%;   //图片添加圆角

      background-color: #eeeeee;

    }

  }

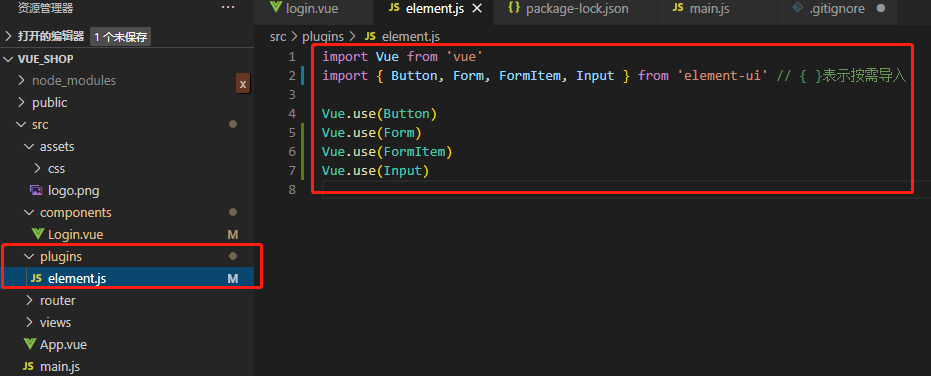
}

### 添加和设置输入框、按钮的位置和样式

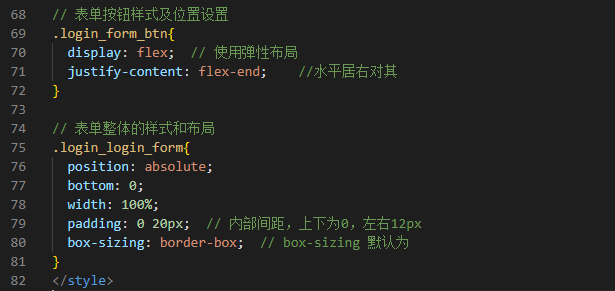
1. 登录页面的布局



1. 上element-ui官网查看相应组件用法
2. 由于项目需要按需导入element UI的组件，需要在plugins/element.js下按需添加需要使用的组件。



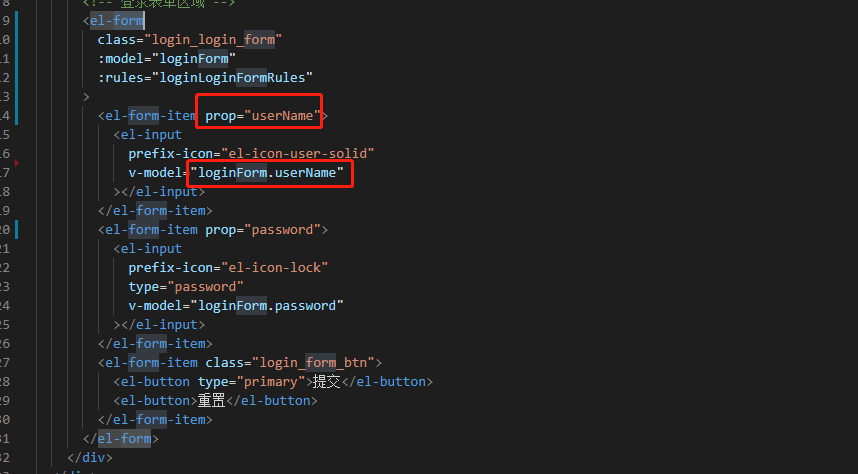
1. 按钮使用flex布局，参考<https://www.runoob.com/w3cnote/flex-grammar.html>学习

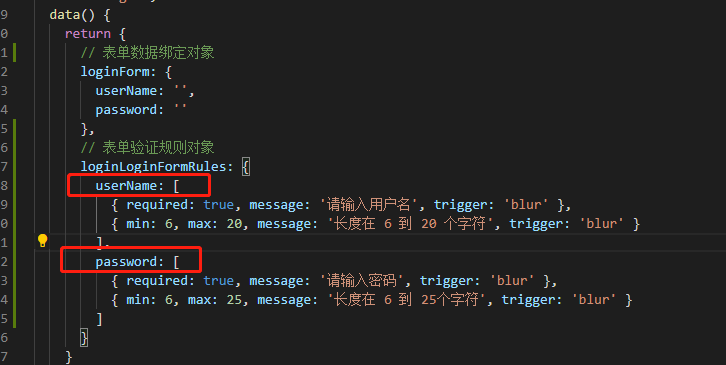


1. Input的样式设置

### 登录框表单数据验证

1. 为el-form 绑定属性 rules 校验对象
2. 在 data()函数中定义校验对象，rules对象的每一个属性都是一个验证规则。
3. 不同的表单项，通过prop制定不同的验证规则





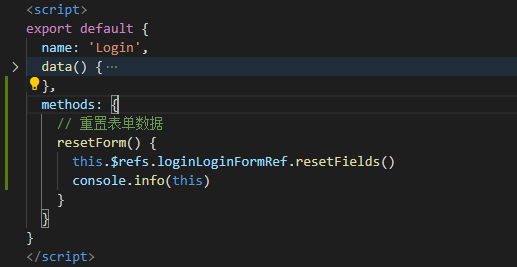
#### 经验教训：

#### 规则的命名 例如userName, 必须要和model的名字loginForm.userName,一样，都叫userName, 否则功能会异常。

### 登录表单数据重置

1. 为表单el-form 添加ref属性。
2. 在.vue组件中，this指代整个组件，通过this.$refs获取到所有的引用，然后

this.$refs.xxxref获得el-form实例，然后调用form表单实例的resetFileds 方法重置表单数据。

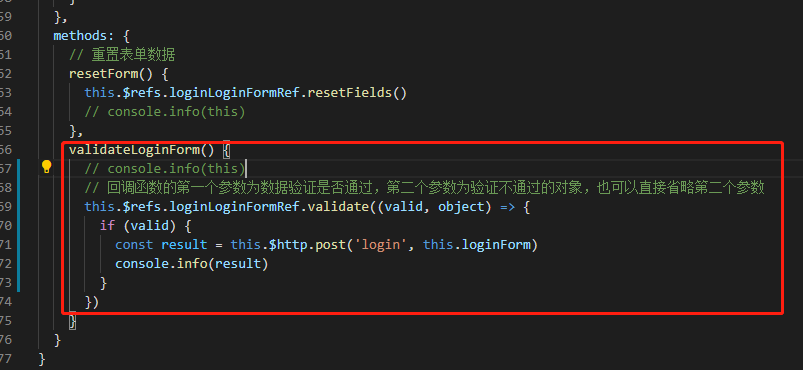


### 登录前预验证

在正式发起登录请求之前，要对表单数据进行整体的预验证，如果表单数据都符合我们定义的rules,则可以发起登录请求，否则提示用户输入数据有误。根据element-ui的文档，resetFileds方法可以进行表单的重置，validate方法对表单进行校验。

1. 拿到表单的引用对象，参考表单数据重置
2. 写回调函数处理validate的结果。如果验证没有问题，则发起真正的网络请求登录验证



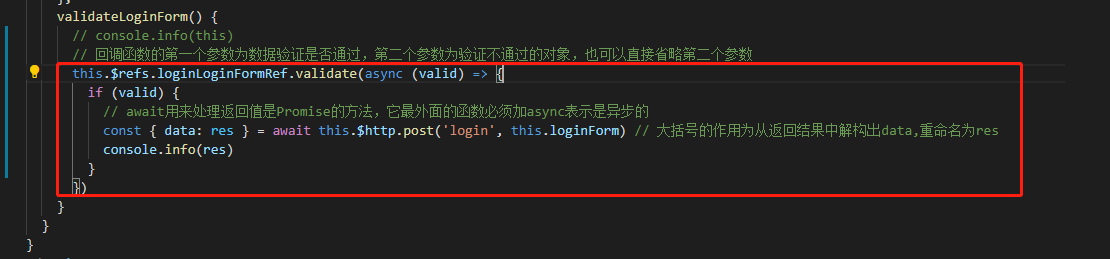


### 配置axios发起登录请求

1. 先导入axios依赖
2. 将axios挂在到vue的原型对象上,这样每一个Vue组件都可以通过this直接访问到 $http



1. 发起请求，并处理请求数据



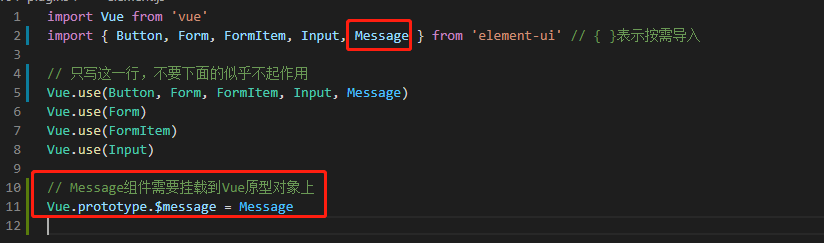
### 使用axios请求拦截器

确保之后每次的请求头中Authorization字段中包含token字段

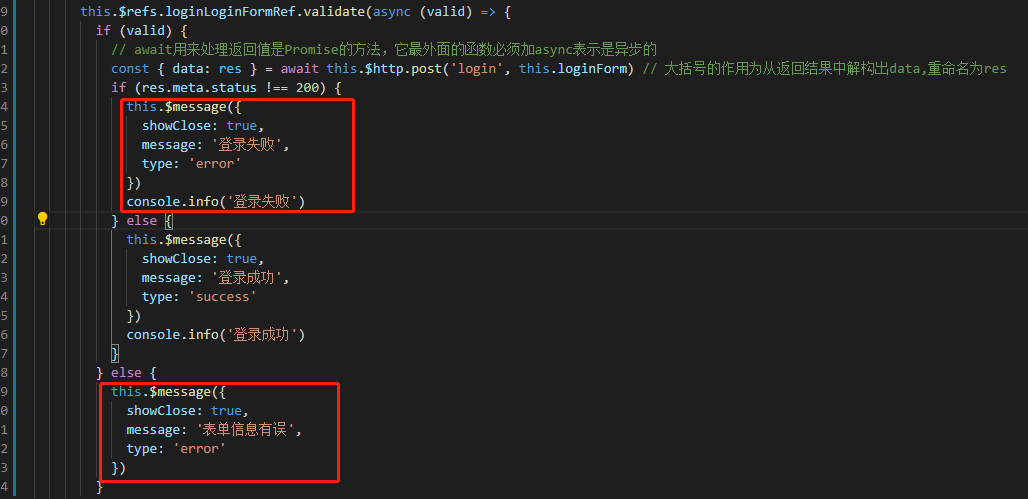


### 使用Element-UI message组件

1. 在element.js 中导入 Message组件
2. 将Message组件挂在到Vue的原型对象上

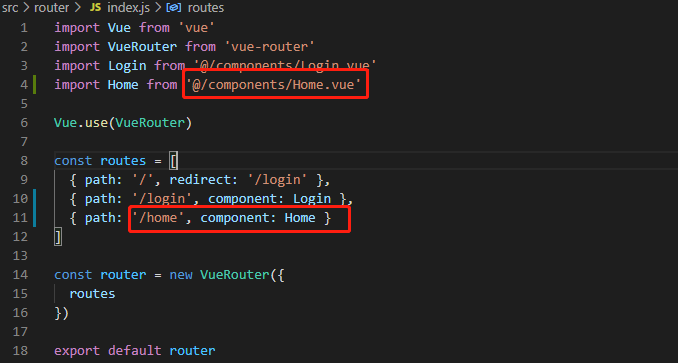


1. 使用Message组件发送消息



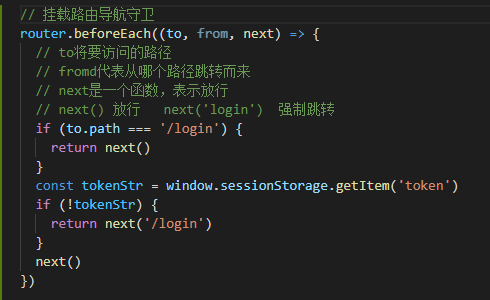
### 保存登录的token及登录成功后的页面跳转

1. 将登录成功后服务器返回数据的token保存到sessionStorage中，
2. 通过编程式导航，将页面跳转到项目主页，/home
3. 注意，@符号后的 /经常忘记写导致找不到组件



### 路由导航守卫

1. 改造router目录下，index.js的代码，添加路由导航首位，只有token的值不为空时，才能进去home页面
2. BeforeEach即为导航守卫



To为将要访问的页面，from表示从哪个页面跳转而来，next()为放行函数。

### 登出功能

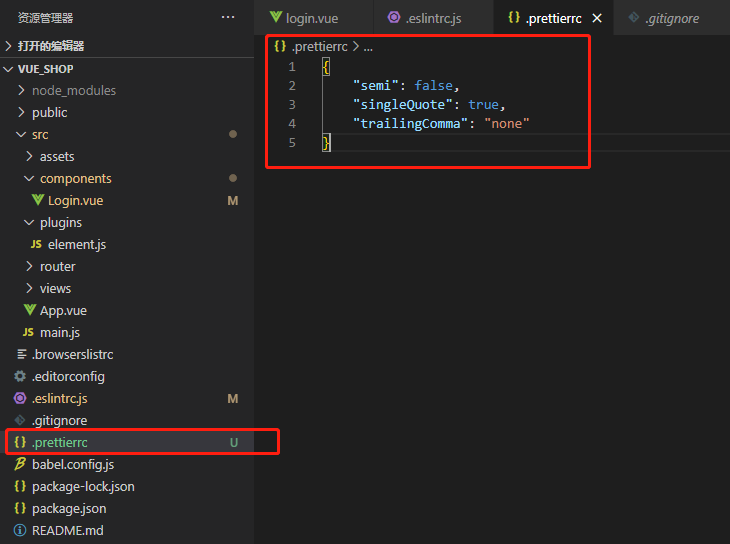
1. 清除sessionStorage
2. 跳转到登录页面



## 如何进行代码格式化的配置

### 解决vscode格式化（prettier）问题

1. 在项目中创建 .prettierrc文件（格式化文件的配置项）



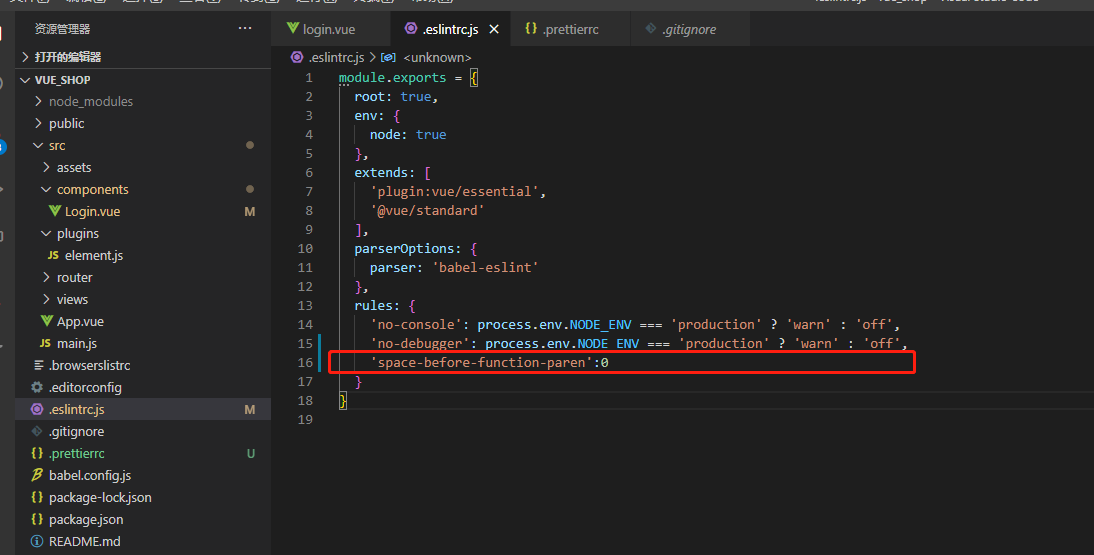
第一项表示，格式化时不加分号

第二项表示格式化代码使用单引号

第三项表示格式化代码不加逗号

### 解决eslint必须要在函数名称与（）之间加空格的问题

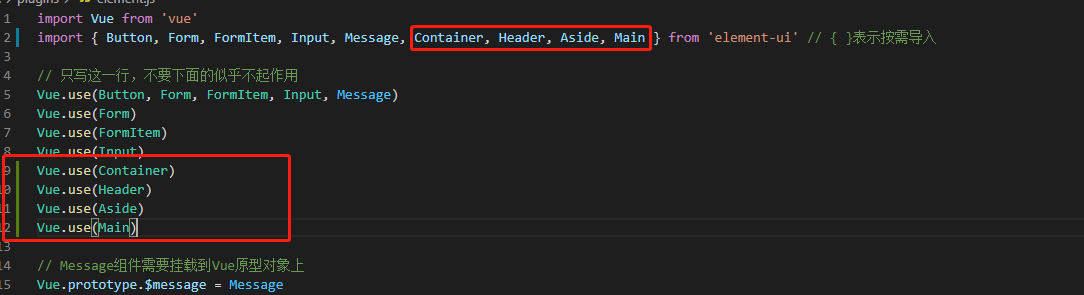
如：data () {},需要修改 .eslintrc.js,将该条规则关闭，值设为0，重启后生效



## 项目主页布局

### 借用element-ui中的布局组件，进行项目主页布局

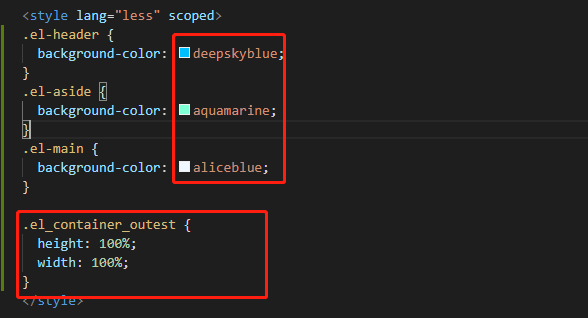
1. 在element.js中依次倒入 Container,Main,ASide,Header



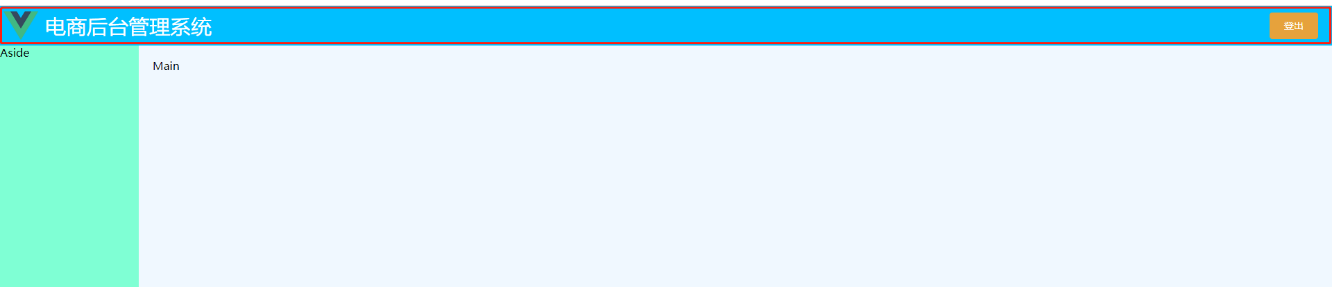
1. 在页面中依次添加各个元素，实现页面布局



1. 通过设置css, 将不同区域设为不同的背景颜色，并调整其大小



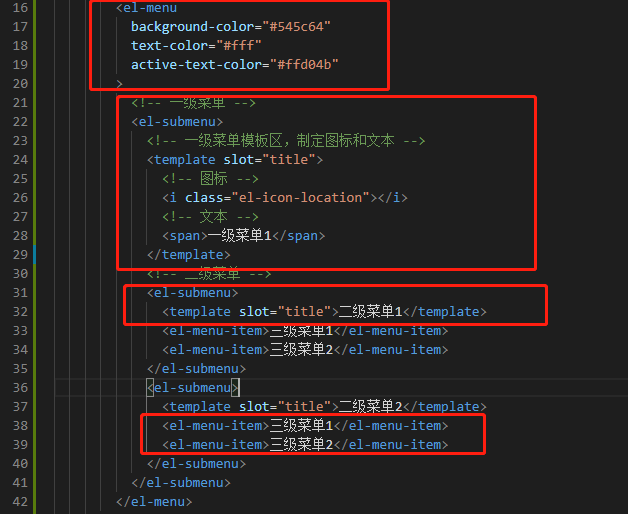
### 实现主页中，Header区域的布局

1. 要实现左右布局，最简单的方法，使用flex布局
2. <style lang="less" scoped>
3. .el-header {
4. background-color: deepskyblue;
5. // 使用flex布局
6. display: flex;
7. justify-content: space-between;
8. padding-left: 5px;
9. align-items: center;
10. color: #ffffff; // 字体颜色，白色
11. font-size: 30px;
12. img {
13. //注意这里img前不加 “.”
14. width: 50px;
15. height: 50px;
16. }
17. > div {
18. display: flex;
19. align-items: center; //居中对齐
20. span {
21. margin-left: 10px; //文本与图片之间的间距
22. }
23. }
24. }
25. .el-aside {
26. background-color: aquamarine;
27. }
28. .el-main {
29. background-color: aliceblue;
30. }
31. .el\_container\_outest {
32. height: 100%;
33. width: 100%;
34. }
35. </style>

## 七、实现左侧菜单栏

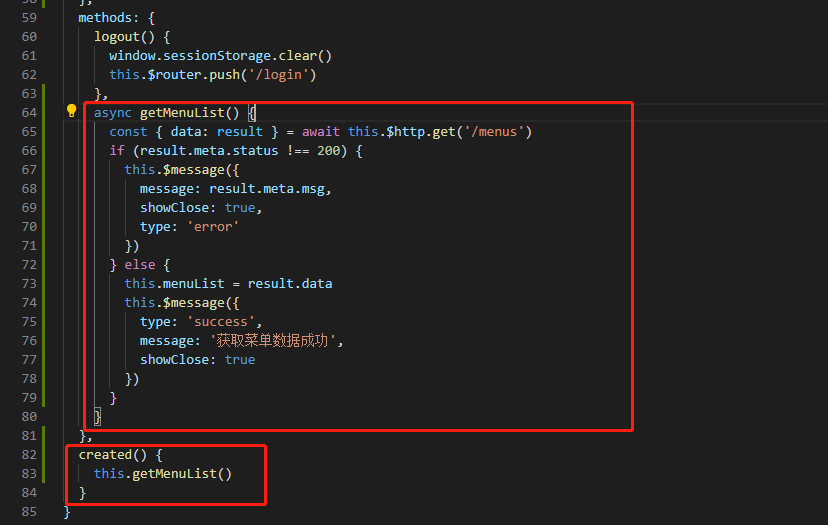
### 菜单项的基本使用

使用el-menu,el-submenu,el-menu-item, 如果一个菜单项包含子菜单，则使用el-submenu包含其子菜单，否则使用el-meue-item。

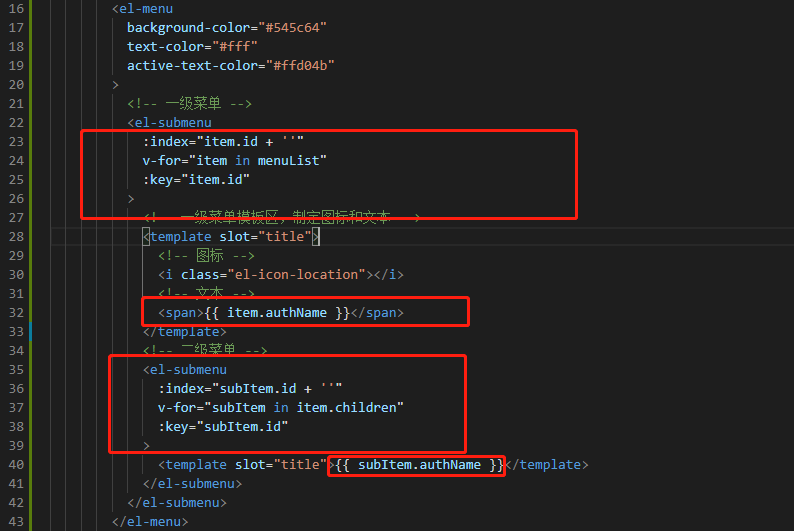


### 获取菜单数据

通过声明周期函数created()，在home页面创建时请求服务器得到菜单数据



### 通过 v-for指令，两层循环渲染左侧菜单数据



注意index为必须的属性，其值需为字符串类型，而返回的数据类型为int,因此加 ‘’转换为字符串类型。

### 选中项颜色&以及菜单的图标

选中项颜色默认就是蓝色，也可以通过active-text-color来指定。只有el-menu-item有颜色，el-submenu没有

动态绑定菜单的图标



### 每次只打开一个以及菜单，并且优化菜单项点开时，右边凸出去的问题

