**VueJava全栈开发**

**目 录**

[第1章 Vue-cli使用 2](#_Toc55166303)

[1.1 安装Node.js(npm) 2](#_Toc55166304)

[1.2 安装vue-cli脚手架 2](#_Toc55166305)

[1.3 使用vue-cli创建项目 2](#_Toc55166306)

[1.3.1 命令行方式创建 2](#_Toc55166307)

[1.3.2 图形界面方式创建 2](#_Toc55166308)

[1.3.3 创建vue.config.js(可选) 8](#_Toc55166309)

[1.4 安装及导入Element-UI 9](#_Toc55166310)

[第2章 静态页开发 9](#_Toc55166311)

[2.1 登录页布局 9](#_Toc55166312)

[2.1.1 创建并引入全局样式 9](#_Toc55166313)

[2.1.2 关闭eslint 10](#_Toc55166314)

[2.1.3 安装less-loader 10](#_Toc55166315)

[2.1.4 创建简单登录页 10](#_Toc55166316)

[2.1.5 删除冗余页面及路由配置 10](#_Toc55166317)

[2.1.6 完善登录页 11](#_Toc55166318)

[2.2 主页布局 15](#_Toc55166319)

[2.2.1 创建主页及路由 15](#_Toc55166320)

[2.2.2 修改登陆页登录事件 16](#_Toc55166321)

[2.2.3 设置主页框架 16](#_Toc55166322)

[2.2.4 主页头部布局 18](#_Toc55166323)

[2.2.5 侧边栏菜单 21](#_Toc55166324)

[2.3 菜单树组件化 25](#_Toc55166325)

# Vue-cli使用

## 安装Node.js(npm)

下载地址 https://nodejs.org/zh-cn/download

下载完毕后安装成功后验证：在命令行输入node –v显示版本

## 安装vue-cli脚手架

npm install --registry=https://registry.npm.taobao.org -g @vue/cli

* *如果已经安装了旧版本，必须先删除它，删除方式如下：*

*npm uninstall vue-cli –g 或 yarn global remove vue-cli*

*参考https://cli.vuejs.org/zh/guide/installation.html*

* *如果嫌每次都要输registry代理地址麻烦，可以配置*

*1).安装cnpm*

*npm install -g cnpm --registry=https://registry.npm.taobao.org*

*2).配置registry和disturl*

*npm config set registry* <https://registry.npm.taobao.org>

*npm config set disturl* https://npm.taobao.org/dist

*或者用npm config edit打开*~/.npmrc自行添加*registry和disturl*

*3).查看修改结果*

*npm config list*

安装成功后验证版本：在命令行输入vue -V或vue --version

## 使用vue-cli创建项目

首先安装git bash，安装完成后命令行设置：

git config --list

git config --global -e

# 提交检出均不转换换行符

git config --global core.autocrlf false

# 拒绝提交包含混合换行符的文件

git config --global core.safecrlf true

### 命令行方式创建

1) 先查看创建帮助参数

vue create --help

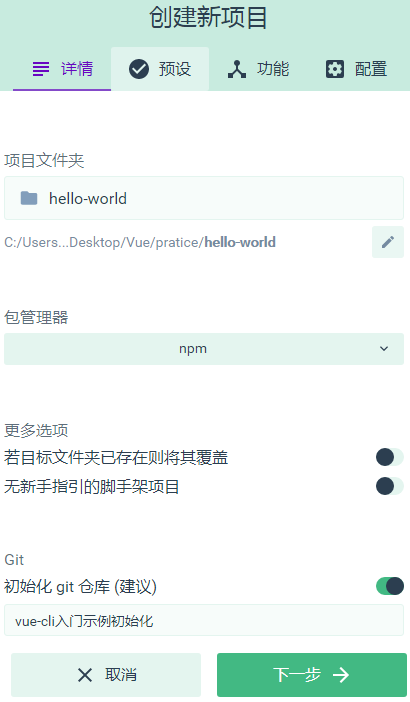
2) 创建项目

vue create 项目名称

### 图形界面方式创建

命令行中输入vue ui会启动一个web图形界面然后根据向导一步步创建。

#### 确定项目名和包管理器



#### 选择如何配置项目插件

选择“手动配置项目”



#### 选择插件

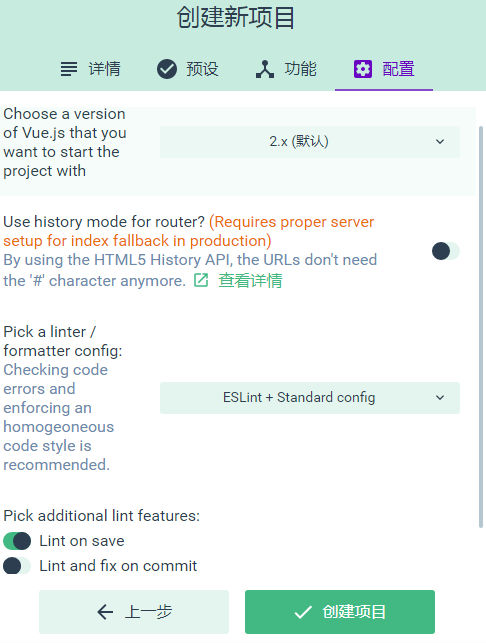
勾选以下选项包括：

* Choose Vue version
* Babel
* Router
* Vuex
* Linter/Formatter
* 使用用配置文件



#### 插件详细配置

选择Vue2.x版本、ESLint + Standard Config



#### 是否保留以上预设

点击“创建项目，不保存预设”按钮

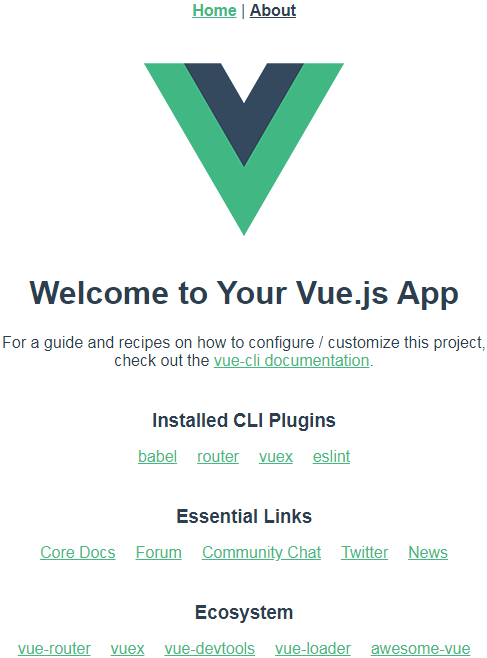


#### 创建结果

默认将启动8000端口并启动仪表盘，地址：http://localhost:8000/dashboard



在仪表盘点击左侧 任务|serve然后点击启动按钮启动项目，项目默认会以8080端口启动，访问地址：http://localhost:8080/#/



### 创建vue.config.js(可选)

在项目目录下创建vue.config.js，内容如下：

// vue.config.js  
***module***.exports = {  
 // 选项...   
}

具体可参考 https://cli.vuejs.org/zh/config/，详细配置可参考如下文件：



## 安装及导入Element-UI

* 停止vue ui的运行
* 将项目导入到webstorm/idea或vscode中
* 切换到项目目录命令行运行 npm i element-ui -S 安装elementui
* 修改src/main.js文件添加elementui的导入

import ElementUI from 'element-ui'  
import 'element-ui/lib/theme-chalk/index.css'  
  
Vue.use(ElementUI)

* 在项目目录命令行运行npm run serve 以开发模式运行程序

# 静态页开发

## 登录页布局

### 创建并引入全局样式

在src/assets/目录下新建css目录，然后在css目录新建global.css样式文件，内容如下：

html,body,#app{  
 height: 100%;  
 padding: 0;  
 margin: 0;  
}

在src/main.js中引入它，内容如下：

import Vue from 'vue'  
import App from './App.vue'  
import ***router*** from './router'  
import store from './store'  
  
import ElementUI from 'element-ui'  
import 'element-ui/lib/theme-chalk/index.css'  
  
import './assets/css/global.css'

Vue.use(ElementUI)  
  
Vue.config.productionTip = false  
  
new Vue({  
 ***router***,  
 store,  
 render: h => h(App)  
}).$mount('#app')

### 关闭eslint

vue.config.js选项内容中添加一行 lintOnSave: false, 结果如下：

// vue.config.js  
***module***.exports = {  
 // 选项...  
 // 关闭eslint  
 lintOnSave: false,  
}

### 安装less-loader

在项目目录下命令行运行

npm i less-loader less -D

### 创建简单登录页

在src/views/目录下新建登录页Login.vue，内容如下：

<template>  
 <!-- 登录容器 -->  
 <div class="login-container">  
 <!-- 登录区域 -->  
 <div class="login-box">  
 <!-- 头像 -->  
 <div class="avatar-box">  
 <img src="../assets/img/avatar.gif" class="avatar-img"/>  
 </div>  
 <!-- 登录表单 -->  
   
 </div>  
 </div>  
</template>  
  
<script>  
 export default {  
 name: 'Login'  
 }  
</script>  
  
<style lang="less" scoped>  
  
</style>

### 删除冗余页面及路由配置

删除src/views/目录下的Home.vue、about.vue

修改src/App.vue删除所有route-link节点，保留<router-view/>元素，内容为：

<template>  
 <div id="app">  
 <router-view/>  
 </div>  
</template>  
  
<style>  
</style>

修改src/router/index.js，内容如下：

import Vue from 'vue'  
import VueRouter from 'vue-router'  
  
Vue.use(VueRouter)  
  
const routes = [  
 {  
 path: '/',  
 redirect: '/login'  
 },  
 {  
 path: '/login',  
 name: 'Login',  
 component: () => import('../views/Login.vue')  
 }  
]  
  
const ***router*** = new VueRouter({  
 routes  
})  
  
export default ***router***

### 完善登录页

修改登录页，参考https://element.eleme.cn/#/zh-CN/component/form，结果如下：

<template>  
 <!-- 登录容器 -->  
 <div class="login-container">  
 <!-- 登录区域 -->  
 <div class="login-box">  
 <!-- 头像 -->  
 <div class="avatar-box">  
 <img src="../assets/img/avatar.gif" class="avatar-img"/>  
 </div>  
 <!-- 登录表单 -->  
 <el-form :model="loginForm" :rules="loginRules" ref="loginForm" label-width="0px" class="login-form">  
 <el-form-item prop="username">  
 <el-input v-model="loginForm.username" prefix-icon="el-icon-user"></el-input>  
 </el-form-item>  
 <el-form-item prop="password">  
 <el-input v-model="loginForm.password" prefix-icon="el-icon-lock"></el-input>  
 </el-form-item>  
 <el-form-item prop="captcha" class="captcha-box">  
 <div class="captcha-box">  
 <el-input v-model="loginForm.captcha" placeholder="输入图像验证码" class="captcha" prefix-icon="el-icon-mobile">  
 </el-input>  
 <img src="../assets/img/captcha.png" alt="" class="captcha-img">  
 </div>  
 </el-form-item>  
 <el-form-item class="login-button">  
 <el-button type="primary" @click="submitForm('loginForm')">立即创建</el-button>  
 <el-button @click="resetForm('loginForm')">重置</el-button>  
 </el-form-item>  
  
 </el-form>  
 </div>  
 </div>  
</template>  
  
<script>  
 export default {  
 name: 'Login',  
 data() {  
 return {  
 loginForm: {  
 username: '',  
 password: '',  
 captcha: '',  
 },  
 loginRules: {  
 username: [  
 { required: true, message: '请输入用户名', trigger: 'blur' },  
 { min: 3, max: 5, message: '长度在 3 到 5 个字符', trigger: 'blur' },  
 ],  
 password: [  
 { required: true, message: '请输入密码', trigger: 'blur' },  
 { min: 3, max: 5, message: '长度在 3 到 5 个字符', trigger: 'blur' },  
 ],  
 captcha: [  
 { required: true, message: '请输入图像验证码', trigger: 'blur' },  
 { min: 3, max: 5, message: '长度在 3 到 5 个字符', trigger: 'blur' },  
 ],  
 }  
 };  
 },  
 methods: {  
 // 提交表单  
 submitForm(formName) {  
 this.$refs[formName].validate((valid) => {  
 if (valid) {  
 alert('submit!');  
 } else {  
 ***console***.log('error submit!!');  
 return false;  
 }  
 });  
 },  
 // 重置表单  
 resetForm(formName) {  
 this.$refs[formName].resetFields();  
 }  
 }  
 }  
</script>  
  
<style lang="less" scoped>  
 .login-container {  
 height: 100%;  
 background-color: #0ff;  
  
 .login-box {  
 width: 320px;  
 height: 350px;  
 position: absolute;  
 top: 50%;  
 left: 50%;  
 margin-left: -160px;  
 margin-top: -175px;  
 background-color: #fff;  
  
 .avatar-box{  
 width: 130px;  
 height: 130px;  
 border: 1px solid #eee;  
 border-radius: 50%;  
 margin: -71px auto;  
 background-color: #fff;  
 padding: 5px;  
  
 .avatar-img{  
 width: 100%;  
 height: 100%;  
 border-radius: 50%;  
 background-color: #eee;  
 }  
 }  
  
 .login-form {  
 position: absolute;  
 bottom: 0;  
 width: 100%;  
 padding: 0 20px;  
 box-sizing: border-box;  
  
 .captcha-box {  
 display: flex;  
 .captcha {  
 width: 70%;  
 justify-content: left;  
 margin-right: 10px;  
 }  
 .captcha-img {  
 width: 30%;  
 justify-content: flex-end;  
 }  
 }  
  
 .login-button {  
 display: flex;  
 justify-content: flex-end;  
 }  
 }  
 }  
 }  
</style>

登陆页运行效果如下图：



## 主页布局

### 创建主页及路由

主页面src/views/Main.vue

<template>  
 <div>这是主页面</div>  
</template>  
  
<script>  
 export default {  
 name: 'Main'  
 }  
</script>  
  
<style scoped>  
  
</style>

添加主页面路由，src/router/index.js在routers数组中添加如下内容：

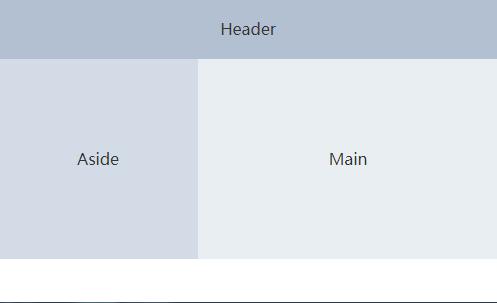
{  
 path: '/main',  
 name: 'Main',  
 component: () => import('../views/Main.vue')  
},

### 修改登陆页登录事件

注释掉alert(‘submit’);添加跳转到主页的方法

// 提交表单  
submitForm(formName) {  
 this.$refs[formName].validate((valid) => {  
 if (valid) {  
 //alert('submit!');  
 this.$router.push("/main");  
 } else {  
 ***console***.log('error submit!!');  
 return false;  
 }  
 });  
},

登录后显示效果如下：

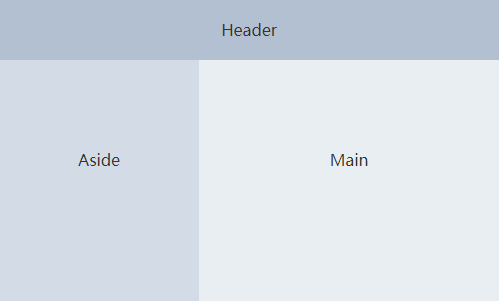


### 设置主页框架

修改src/views/Main.vue，参考https://element.eleme.cn/#/zh-CN/component/container，设置主容器高度100% 内容如下：

<template>  
 <el-container class="main-container">  
 <el-header>Header</el-header>  
 <el-container>  
 <el-aside width="200px">Aside</el-aside>  
 <el-main>Main</el-main>  
 </el-container>  
 </el-container>  
</template>  
  
<script>  
 export default {  
 name: 'Main'  
 }  
</script>  
  
<style scoped>  
 .main-container {  
 height: 100%;  
 }  
  
 .el-header, .el-footer {  
 background-color: #B3C0D1;  
 color: #333;  
 text-align: center;  
 line-height: 60px;  
 }  
  
 .el-aside {  
 background-color: #D3DCE6;  
 color: #333;  
 text-align: center;  
 line-height: 200px;  
 }  
  
 .el-main {  
 background-color: #E9EEF3;  
 color: #333;  
 text-align: center;  
 line-height: 160px;  
 }  
  
 body > .el-container {  
 margin-bottom: 40px;  
 }  
  
</style>

登录后显示效果如下：



### 主页头部布局

参考https://element.eleme.cn/#/zh-CN/component/dropdown，修改主页后如下：

<template>  
 <!--主容器-->  
 <el-container class="main-container">  
 <el-header>  
 <!-- logo 和项目名称 -->  
 <div class="left-box">  
 <img src="../assets/img/logo.gif" alt=""/>  
 <span>后台管理系统</span>  
 </div>  
 <!-- 登录显示的头像-->  
 <div class="right-box">  
 <el-dropdown trigger="click">  
  
 <span class="el-dropdown-link"><img src="../assets/img/avatar.gif" alt=""/>  
 <i class="el-icon-arrow-down el-icon--right"></i>  
 </span>  
 <!-- 下拉菜单 -->  
 <el-dropdown-menu slot="dropdown">  
 <el-dropdown-item icon="el-icon-plus">黄金糕</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-plus">狮子头</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-plus-outline">螺蛳粉</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-check">双皮奶</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-check">蚵仔煎</el-dropdown-item>  
 </el-dropdown-menu>  
 </el-dropdown>  
 </div>  
 </el-header>  
 <el-container>  
 <el-aside width="200px">Aside</el-aside>  
 <el-main>Main</el-main>  
 </el-container>  
 </el-container>  
</template>  
  
<script>  
 export default {  
 name: 'Main'  
 }  
</script>  
  
<style lang="less" scoped>  
 /\*主容器\*/  
 .main-container {  
 height: 100%;  
 }  
  
 /\*头部\*/  
 .el-header {  
 background-color: #fff;  
 display: flex;  
 justify-content: space-between;  
 color: #fff;  
 padding-left: 0;  
 align-items: center;  
 font-size: 20px;  
  
 /\*左侧log 和 标题\*/  
 .left-box {  
 display: flex;  
 align-items: center;  
  
 /\*logo\*/  
 img{  
 width: 60px;  
 height: 60px;  
 /\*margin-bottom: 10px;\*/  
 }  
  
 /\*标题\*/  
 span {  
 margin-left: 15px;  
 color: #000;  
 }  
 }  
  
 /\*右边的登录头像和菜单\*/  
 .right-box {  
 /\*登录头像\*/  
 .el-dropdown-link > img {  
 width: 60px;  
 border-radius: 50%;  
 background-color: #000;  
 margin-bottom: -5px;  
 }  
 }  
 }  
  
 /\*侧边栏\*/  
 .el-aside {  
 background-color: #D3DCE6;  
 color: #333;  
 /\*text-align: center;\*/  
 line-height: 200px;  
 }  
  
 /\*主题内容\*/  
 .el-main {  
 background-color: #E9EEF3;  
 color: #333;  
 text-align: center;  
 line-height: 160px;  
 }  
  
 .el-dropdown-link {  
 cursor: pointer;  
 }  
</style>

显示效果如下：



### 侧边栏菜单

参考https://element.eleme.cn/#/zh-CN/component/menu，具体代码如下：

<template>  
 <!--主容器-->  
 <el-container class="main-container">  
 <el-header>  
 <!-- logo 和项目名称 -->  
 <div class="left-box">  
 <img src="../assets/img/logo.gif" alt=""/>  
 <span>后台管理系统</span>  
 </div>  
 <!-- 登录显示的头像-->  
 <div class="right-box">  
 <el-dropdown trigger="click">  
  
 <span class="el-dropdown-link"><img src="../assets/img/avatar.gif" alt=""/>  
 <i class="el-icon-arrow-down el-icon--right"></i>  
 </span>  
 <!-- 下拉菜单 -->  
 <el-dropdown-menu slot="dropdown">  
 <el-dropdown-item icon="el-icon-plus">黄金糕</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-plus">狮子头</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-plus-outline">螺蛳粉</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-check">双皮奶</el-dropdown-item>  
 <el-dropdown-item icon="el-icon-circle-check">蚵仔煎</el-dropdown-item>  
 </el-dropdown-menu>  
 </el-dropdown>  
 </div>  
 </el-header>  
 <el-container>  
 <el-aside :width="isCollapse?'64px':'200px'">  
 <div class="toggle-box" @click="toggleCollapse">|||</div>  
 <el-menu :default-active="activeIndex" mode="vertical" @select="handleSelect" background-color="#545c64" text-color="#fff" active-text-color="#ffd04b"  
 :collapse="isCollapse"  
 :collapse-transition="false"  
 :unique-opened="true">  
 <el-menu-item index="1"><i class="el-icon-document"></i><span>消息中心</span></el-menu-item>  
 <el-submenu index="2">  
 <template slot="title"><i class="el-icon-setting"></i><span>系统管理</span></template>  
 <el-menu-item index="2-1"><i class="el-icon-user-solid"></i><span>角色管理</span></el-menu-item>  
 <el-menu-item index="2-2"><i class="el-icon-user"></i><span>用户管理</span></el-menu-item>  
 <el-menu-item index="2-3"><i class="el-icon-menu"></i><span>菜单管理</span></el-menu-item>  
 <el-submenu index="2-4">  
 <template slot="title"><i class=""></i>选项4</template>  
 <el-menu-item index="2-4-1"><span>选项1</span></el-menu-item>  
 <el-menu-item index="2-4-2"><span>选项2</span></el-menu-item>  
 <el-menu-item index="2-4-3"><span>选项3</span></el-menu-item>  
 </el-submenu>  
 </el-submenu>  
 </el-menu>  
 </el-aside>  
 <el-main>Main</el-main>  
 </el-container>  
 </el-container>  
</template>  
  
<script>  
 export default {  
 name: 'Main',  
 data() {  
 return {  
 activeIndex: '1',  
 isCollapse: false,  
 };  
 },  
 methods: {  
 handleSelect(key, keyPath) {  
 ***console***.log(key, keyPath);  
 },  
 toggleCollapse() {  
 this.isCollapse = !this.isCollapse;  
 }  
 }  
 }  
</script>  
  
<style lang="less" scoped>  
 /\*主容器\*/  
 .main-container {  
 height: 100%;  
 }  
  
 /\*头部\*/  
 .el-header {  
 background-color: #fff;  
 display: flex;  
 justify-content: space-between;  
 color: #fff;  
 padding-left: 0;  
 align-items: center;  
 font-size: 20px;  
  
 /\*左侧log 和 标题\*/  
 .left-box {  
 display: flex;  
 align-items: center;  
  
 /\*logo\*/  
 img{  
 width: 60px;  
 height: 60px;  
 }  
  
 /\*标题\*/  
 span {  
 margin-left: 15px;  
 color: #000;  
 }  
 }  
  
 /\*右边的登录头像和菜单\*/  
 .right-box {  
 /\*登录头像\*/  
 .el-dropdown-link > img {  
 width: 60px;  
 border-radius: 50%;  
 background-color: #000;  
 margin-bottom: -5px;  
 }  
 }  
 }  
  
 /\*侧边栏\*/  
 .el-aside {  
 background-color: #545c64;  
 color: #333;  
 /\*text-align: center;\*/  
 line-height: 200px;  
  
 /\*去掉边框\*/  
 .el-menu {border: none;}  
  
 .toggle-box {  
 background-color: #545c64;  
 font-size: 20px;  
 font-weight: bold;  
 line-height: 20px;  
 color: #fff;  
 letter-spacing: .2em;  
 text-align: center;  
 cursor: pointer;  
 }  
 }  
  
 /\*主题内容\*/  
 .el-main {  
 background-color: #E9EEF3;  
 color: #333;  
 text-align: center;  
 line-height: 160px;  
 }  
  
 .el-dropdown-link {  
 cursor: pointer;  
 }  
</style>

运行效果如下：



## 菜单树组件化

在src/components/目录下新建菜单组件MenuTree.vue，拷贝Main.vue中el-menu里面的内容到MenuTree.vue的<template>元素中，外部再套一层div，如下：

<template>  
 <div>  
 <template v-for="***item*** in this.menuList">  
 <!-- 循环菜单json列表生成（如上格式）菜单 -->  
 <el-submenu :disabled="***item***.disabled" :index="***item***.id+''" :key="***item***.id+''" v-if="***item***.children.length>0">  
 <template slot="title">  
 <i :class="***item***.icon" v-if="***item***.icon"></i><span>{{***item***.menuName}}</span>  
 </template>  
 <MenuTree :menuList="***item***.children"></MenuTree>  
 </el-submenu>  
 <el-menu-item :disabled="***item***.disabled" :index="***item***.id+''" :key="***item***.id" :route="***item***.url+''" @click="savePath(***item***.url)" v-else>  
 <i :class="***item***.icon" v-if="***item***.icon"></i><span>{{***item***.menuName}}</span>  
 </el-menu-item>  
 </template>  
 </div>  
</template>  
  
<script>  
 export default {  
 name: 'MenuTree',  
 data() {  
 return {  
  
 }  
 },  
 // 接收外部传递过来的参数  
 props: ["menuList"],

// 保存激活路径  
 savePath(path){  
 ***window***.sessionStorage.setItem("activePath", path);  
 // 赋值给父组件的activePath属性  
 this.activePath = path;  
 }

}  
</script>  
  
<style scoped>  
 /\* 实现了溢出处理，在菜单收缩时使文字隐藏\*/  
 .el-menu--collapse span,  
 .el-menu--collapse i.el-submenu\_icon-arrow {  
 height: 0;  
 width: 0;  
 overflow: hidden;  
 visibility: hidden;  
 display: inline-block;  
 }  
</style>

在Main.vue中使用MenuTree组件，在Main.vue中导入并注册组件：

<script>  
 // 导入组件  
 import MenuTree from '../components/MenuTree'  
 export default {  
 name: 'Main',  
 // 注册组件  
 components: { MenuTree },

data() {  
 return {  
 isCollapse: false,

activePath: '',  
 menuList:[],

}  
 //...  
</script>

菜单列表对象menuList的测试数据使用ElementUI菜单.json：



Main.vue中使用组件：

<el-menu :default-active="activePath" mode="vertical" @select="handleSelect" background-color="#545c64" text-color="#fff" active-text-color="#ffd04b"  
 :collapse="isCollapse"  
 :collapse-transition="false"  
 :unique-opened="true" :router="true">  
 <MenuTree :menuList="this.menuList"></MenuTree>  
</el-menu>

## 开发用户管理页面

### 修改主体内容

修改Main.vue的.el-main样式

/\*主体内容\*/  
.el-main {  
 background-color: #E9EEF3;  
 color: #333;  
 text-align: center;  
 line-height: 160px;  
}

改为：

/\*主体内容\*/  
.el-main {  
 background-color: #E9EEF3;   
}

#### 添加路由视图

修改Main.vue的e-main节点为:

<el-main>  
 <!-- 路由视图 -->  
 <router-view></router-view>  
</el-main>

#### 调整框架页样式

删除主题内容样式.el-main内部的text-align: center;和line-height: 160px;，或者注释掉它们：

/\*主体内容\*/  
.el-main {  
 background-color: #E9EEF3;  
 color: #333;  
 /\*text-align: center;\*/  
 /\*line-height: 160px;\*/  
}

### 创建用户管理页面

创建src/views/admin/Admin.vue页面，内容为：

<template>  
 <div>用户管理</div>  
</template>  
  
<script>  
 export default {  
 name: 'Admin'  
 }  
</script>  
  
<style scoped>  
  
</style>

### 配置用户管理页面路由

修改src/router/index.js的main视图添加children数组，把admin的路由配置进children：

{  
 path: '/main',  
 name: 'Main',  
 component: () => import('../views/Main.vue'), },

修改后：

{  
 path: '/main',  
 name: 'Main',  
 component: () => import('../views/Main.vue'),  
 children:[  
 {  
 path: '/admin',  
 name: 'Admin',  
 component: () => import('../views/admin/Admin.vue'),  
 }  
 ]  
},

访问http://localhost:8080/#/admin运行结果：



## 欢迎页

### 使用开源表单设计器及代码生成器

https://mrhj.gitee.io/form-generator/#/