

Vue前端工程化及基于组件的开发

参考资料：

1、Vue2官网：[安装—Vue.js](#)

2、B站黑马程序员[Day03-04. 前端工程化-环境准备哔哩哔哩bilibili](#)

1、什么是Vue前端工程化？

提供如下能力的Vue前端开发环境：

- 统一的目录结构；
- 本地调试；
- 热部署；
- 单元测试；
- 集成打包上线；

Vue-cli是Vue官方提供的一个脚手架，可以快速生成一个具备上述能力的Vue项目模板。

Vue-cli运行的环境：Node.js；

2、如何安装Node.js？

先安装Node.js的版本管理器：nvm.exe；

为什么不直接安装node.js，而要通过nvm来安装？

安装完毕NVM.exe后，再实用nvm.exe来安装指定版本的node.js：

```
1 //比如：安装16.19.1版本的node.js
2 nvm install 16.19.1
3 //安装完成后，显示当前电脑中已经安装的不同版本的node.js：
4 nvm list
5 //指定当前环境中，使用16.19.1版本的node.js
6 nvm use 16.19.1
```

在命令行环境下，测试当前的node.js环境：

```
1 //查看当前node.js的版本：
2 node -v
3 //查看npm的版本：
4 npm -v
```

npm是node.js安装包的命令，类似于ubuntu中的apt。

为了加快node.js安装包的速度，将其使用的安装包的源地址，配置为国内的淘宝，命令如下：

```
1 | npm config set registry https://registry.npm.taobao.org
```

3、安装Vue-cli脚手架

```
1 //安装命令；安装vue  
2 npm install -g @vue/cli
```

确定是否安装成功：

```
1 vue --version
```

如果能看到版本号，就说明脚手架安装成功。

4、创建Vue项目

命令行： vue create vue-project

图形化界面： vue ui

使用图形界面创建一个新的Vue项目；

Vue初始项目文件介绍；

如何运行Vue项目；

```
1 npm run serve
```

vs code中安装Vue的插件： vetur

App.vue组件文件：

```
1 <template>  
2   <div>  
3     {{message}}  
4   </div>  
5 </template>  
6  
7 <script>  
8   export default{  
9     data(){  
10       return{  
11         message:"hello vue"  
12       }  
13     },  
14     methods:{  
15       }  
16     }  
17   }  
18 </script>  
19  
20 <style>  
21   .h{  
22     color:red;  
23   }  
24 </style>
```

5、Element组件库

一套组件库，组件是组成网页的部件，比如超链接，按钮，图片，表格等。

Element - 网站快速成型工具

使用element的好处：快速使用一些效果还不错的组件。

黑马对应视频：[Day03-07. Element-快速入门哔哩哔哩bilibili](#)

Element快速入门案例：

步骤1：安装ElementUI组件库；

```
1 | npm install element-ui@2.15.14
```

步骤2：引入ElementUI组件库：

```
1 | //下列代码添加到main.js中;
2 | import ElementUI from 'element-ui';
3 | import 'element-ui/lib/theme-chalk/index.css';
4 |
5 | Vue.use(ElementUI);
```

步骤3：访问官网，复制组件代码并调整；

```
1 | //比如复制按钮组件;
2 | //https://element.eleme.cn/#/zh-CN/component/button
3 | <el-row>
4 |   <el-button>默认按钮</el-button>
5 |   <el-button type="primary">主要按钮</el-button>
6 |   <el-button type="success">成功按钮</el-button>
7 |   <el-button type="info">信息按钮</el-button>
8 |   <el-button type="warning">警告按钮</el-button>
9 |   <el-button type="danger">危险按钮</el-button>
10 | </el-row>
```

将上述的代码写在：/views/element/ElementView.vue中：

```
1 | <template>
2 | <div>
3 |   <el-row>
4 |     <el-button>默认按钮</el-button>
5 |     <el-button type="primary">主要按钮</el-button>
6 |     <el-button type="success">成功按钮</el-button>
7 |     <el-button type="info">信息按钮</el-button>
8 |     <el-button type="warning">警告按钮</el-button>
9 |     <el-button type="danger">危险按钮</el-button>
10 |   </el-row>
11 | </div>
12 | </template>
13 |
14 | <script>
15 | export default{
16 | }
```

```
17 }
18 </script>
19
20 <style>
21 </style>
```

在App.vue中引入ElementView组件：

```
1 <template>
2   <div>
3     <h1>{{message}}</h1>
4     //引入elementview组件;
5     <element-view>
6     </element-view>
7   </div>
8 </template>
9
10 <script>
11   //import element view组件;
12   import ElementView from './views/element/ElementView.vue'
13   export default{
14     components: { ElementView },
15     data(){
16       return{
17         message:"hello vue"
18       }
19     },
20     methods:{
21
22     }
23   }
24 </script>
25
26 <style>
27   h1{
28     color:red;
29   }
30 </style>
```

6、Element常见组件

table组件：

[组件 | Element](#)

注意复制过程中的自动格式化快捷键：

```
1 1、在windows环境下,ctrl+s调出快捷命令查询框;
2 2、查找format document命令的快捷键: shift + alt + F
```

分页组件：

[组件 | Element](#)

对话框组件：

[组件 | Element](#)

Form组件：

[组件 | Element](#)

7、Element案例

Element布局组件：

[组件 | Element](#)

步骤1：

找到需要的页面布局container，找到目标布局，并复制代码：

```
1 <el-container>
2   <el-header>Header</el-header>
3   <el-container>
4     <el-aside width="200px">Aside</el-aside>
5     <el-main>Main</el-main>
6   </el-container>
7 </el-container>
```

The screenshot shows a user management interface. The sidebar on the left has '系统信息管理' expanded, showing '用户管理' and '题目管理'. The main content area displays a table of users with columns: 日期 (Date), 用户名 (Username), 密码 (Password), and 操作 (Operations). The table data is as follows:

日期	用户名	密码	操作
2016-05-02	王小虎	123456	<button>编辑</button> <button>删除</button>
2016-05-04	王小虎	123456	<button>编辑</button> <button>删除</button>
2016-05-01	王小虎	123456	<button>编辑</button> <button>删除</button>
2016-05-03	王小虎	123456	<button>编辑</button> <button>删除</button>

At the bottom, there is a pagination bar with pages 1 through 100.

header的设置：

对于header，只需要设置下背景颜色，及字体大小即可，具体的代码如下：

```
1 <el-header style="font-size: 40px; background-color: #2e3436; color: white">Quiz
后台管理</el-header>
```

侧边导航栏的设置：

[组件 | Element](#)

采用案例中的菜单栏，再进行调整。

main区域设置：

- 顶部form表单；主要同时复制js部分代码；

[组件 | Element](#)

- 带编辑/删除操作选项的table项目；

[组件 | Element](#)

- 分页选项；

[组件 | Element](#)

添加用户的弹出对话框：

当点击添加用户按钮后，要弹出对话框；

[组件 | Element](#)

除了页面布局以外，还要完成Vue路由设置；

路由多页面设置：

步骤1：安装vue router，一般在新建项目时，已经选择，如果未选择，则以下命令：

```
1 //要使用vue router，要先安装vue router:  
2 npm intall vue-router@3.5.1
```

步骤2：新建组件页面；

步骤3：配置/router/index.js页面：

```
1 {  
2   path: '/user',  
3   name: 'user',  
4   component: ()=>import('../views/element/UserView.vue')  
5 },  
6 {  
7   path: '/question',  
8   name: 'question',  
9   component: ()=>import('../views/element/QuestionView.vue')  
10 },  
11 {  
12   path: '/',  
13   redirect: '/user'  
14 }
```

步骤4：在View页面添加链接选项：

```
1 <el-menu-item index="1-1">  
2   <router-link to="/user">用户管理</router-link>  
3 </el-menu-item>  
4 <el-menu-item index="1-2">  
5   <router-link to="/question">题目管理</router-link>  
6 </el-menu-item>
```

步骤5：修改动态展示区域，在App.vue处修改：

```
1 <router-view></router-view>
```

网页缩放导致的问题



对应的黑马视频：

[Day03-05. 前端工程化-Vue项目哔哩哔哩bilibili](#)