

一、模块化开发

1. vue-router模块化

```
cnpm install vue-router -S
```

1.1 编辑main.js

1.2 编辑App.vue

1.3 编辑router.config.js

2. axios模块化

```
cnpm install axios -S
```

使用axios的两种方式：

方式1：在每个组件中引入axios

方式2：在main.js中全局引入axios并添加到Vue原型中

3. 为自定义组件添加事件

二、Elment UI

1. 简介

Element UI是饿了么团队提供的一套基于Vue2.0的组件库，可以快速搭建网站，提高开发效率

ElementUI PC端

MintUI 移动端

[官网](#)

2. 快速上手

2.1 安装element ui

```
cnpm install element-ui -S
```

2.2 在main.js中引入并使用组件

```
import ElementUI from 'element-ui'
import 'element-ui/lib/theme-default/index.css' //该样式文件需要单独引入
Vue.use(ElementUI);
这种方式引入了ElementUI中所有的组件
```

2.3 在webpack.config.js中添加loader

CSS样式和字体图标都需要由相应的loader来加载，所以需要style-loader、css-loader

默认并没有style-loader模块，所以需要单独安装

```
cnpm install style-loader --save-dev
```

2.4 使用组件

2.5 使用less

安装loader，需要两个：less、less-loader

```
cnpm install less less-loader -D
```

在webpack.config.js中添加loader

3. 按需引入组

3.1 安装babel-plugin-component

```
cnpm install babel-plugin-component -D
```

3.2 配置.babelrc文件

```
"plugins": [["component", [
  {
    "libraryName": "element-ui",
    "styleLibraryName": "theme-default"
  }
]]]
```

3.3 只引入需要的插件

三、自定义全局组件（插件）

全局组件（插件）：就是指可以在main.js中使用Vue.use()进行全局引入，然后在其他组件中就都可以使用了，如

vue-router

```
import VueRouter from 'vue-router'
Vue.use(VueRouter);
```

普通组件（插件）：每次使用时都要引入，如axios

```
import axios from 'axios'
```

四、Vuex

1. 简介

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。

简单来说，用来集中管理数据，类似于React中的Redux，都是基于Flux的前端状态管理框架

2. 基本用法

2.1 安装vuex

```
cnpm install vuex -S
```

2.2 创建store.js文件，在main.js中导入并配置store.选项

2.3 编辑store.js文件

Vuex的核心是Store(仓库)，相当于是一个容器，一个store实例中包含以下属性的方法：

state	定义属性（状态、数据）
getters	用来获取属性
actions	定义方法（动作）
commit	提交变化，修改数据的唯一方式就是显式的提交mutations
mutations	定义变化

注：不能直接修改数据，必须显式提交变化，目的是为了追踪到状态的变化

2.4 编辑App.vue

在子组件中访问store对象的两种方式：

方式1：通过this.\$store访问

方式2：通过mapState、mapGetters、mapActions访问，vuex提供了两个方法：

mapState	获取state
mapGetters	获取getters
mapActions	获取actions

3. 分模块组织Vuex

```
| -src
  | -store
    | -index.js
    | -getters.js
    | -actions.js
    | -mutations.js
    | -modules //分为多个模块，每个模块都可以拥有自己的state、getters、actions、mutations
      | -user.js
      | -cart.js
      | -goods.js
      | ....
```