Nuxt前端搭建

# 一、前端搭建

## 1、服务端渲染技术NUXT

### 1.1 什么是服务端渲染

服务端渲染又称SSR (Server Side Render)是在服务端完成页面的内容，而不是在客户端通过AJAX获取数据。

服务器端渲染(SSR)的优势主要在于：更好的 SEO，由于搜索引擎爬虫抓取工具可以直接查看完全渲染的页面。

如果你的应用程序初始展示 loading 菊花图，然后通过 Ajax 获取内容，抓取工具并不会等待异步完成后再进行页面内容的抓取。也就是说，如果 SEO 对你的站点至关重要，而你的页面又是异步获取内容，则你可能需要服务器端渲染(SSR)解决此问题。

另外，使用服务器端渲染，我们可以获得更快的内容到达时间(time-to-content)，无需等待所有的 JavaScript 都完成下载并执行，产生更好的用户体验，对于那些内容到达时间(time-to-content)与转化率直接相关的应用程序而言，服务器端渲染(SSR)至关重要。

### 1.2 什么是NUXT

Nuxt.js 是一个基于 Vue.js 的轻量级应用框架,可用来创建服务端渲染 (SSR) 应用,也可充当静态站点引擎生成静态站点应用,具有优雅的代码结构分层和热加载等特性。

官网网站：

<https://zh.nuxtjs.org/>

## 2、NUXT环境初始化

### 2.1 下载压缩包

地址：<https://github.com/nuxt-community/starter-template/archive/master.zip>

### 2.2 解压后修改package.json的作者等个性信息

### 2.3 修改nuxt.config.js的个性信息

### 2.4 终端中进入项目目录安装依赖

1. npm install

### 2.5 引入element-ui

1. npm install element-ui

在plugins文件夹下创建**myPlugin.js**文件；

在myPlugin.js文件引入element-ui：

1. **import** Vue from 'vue'
2. **import** ElementUI from 'element-ui' //element-ui的全部组件
3. **import** 'element-ui/lib/theme-chalk/index.css' //element-ui的css
4. Vue.use(ElementUI) //使用elementUI

在nuxt.config.js文件中使用myPlugin.js，在build下面添加内容：

1. plugins: [
2. { src: '~/plugins/myPlugin.js', ssr: **false** }
3. ]

### 2.6 启动

1. npm run dev

访问项目：<http://localhost:3000/>

## 3、封装axios

安装

1. npm install axios

封装

1. **import** axios from 'axios'
2. **import** { MessageBox, Message } from 'element-ui'
3. // 创建axios实例
4. **const** service = axios.create({
5. baseURL: 'http://localhost',
6. timeout: 15000 // 请求超时时间
7. })
8. // http request 拦截器
9. service.interceptors.request.use(
10. config => {
11. // token 先不处理，后续使用时在完善
12. **return** config
13. },
14. err => {
15. **return** Promise.reject(err)
16. })
17. // http response 拦截器
18. service.interceptors.response.use(
19. response => {
20. **if** (response.data.code !== 200) {
21. Message({
22. message: response.data.message,
23. type: 'error',
24. duration: 5 \* 1000
25. })
26. **return** Promise.reject(response.data)
27. } **else** {
28. **return** response.data
29. }
30. },
31. error => {
32. **return** Promise.reject(error.response)
33. })
34. **export** **default** service

## 4、首页实现

### 4.1 添加静态资源

### 4.2 定义布局

### 4.3 提取头/尾文件

### 4.4 默认布局引入头尾文件

### 4.5 首页数据分析及前后端实现

1，获取医院等级（根据数据字典编码获取）

2，获取地区（根据数据字典编码获取）

3，医院分页列表

4，根据医院名称关键字搜索医院列表

## 5、医院详情的前后端实现