**【全栈FootPrints-01】Nuxt.js 搭建网站基础开发框架**

[](https://xiaozhuanlan.com/u/giscafer)

[giscafer](https://xiaozhuanlan.com/u/giscafer)已关注

预计阅读时间20分钟17 天前

本篇为 **旅游足迹网站（FootPrints）** 第1篇，主要介绍从0搭建一个 Nuxt.js 的网站开发基础框架，主要内容包括：

* 确定 Nuxt.js 基础框架中的各种技术的选型
* 工程规范和代码结构了解
* 布局设计与开发

**创建 Nuxt.js 工程 & 技术选型**

Nuxt.js 中文官方站点：<https://zh.nuxtjs.org/guide/installation>

**安装 Nuxt.js**

通过 create-nuxt-app 可以快速创建 Nuxt.js 应用工程，这里我命名工程名为 footprint-app

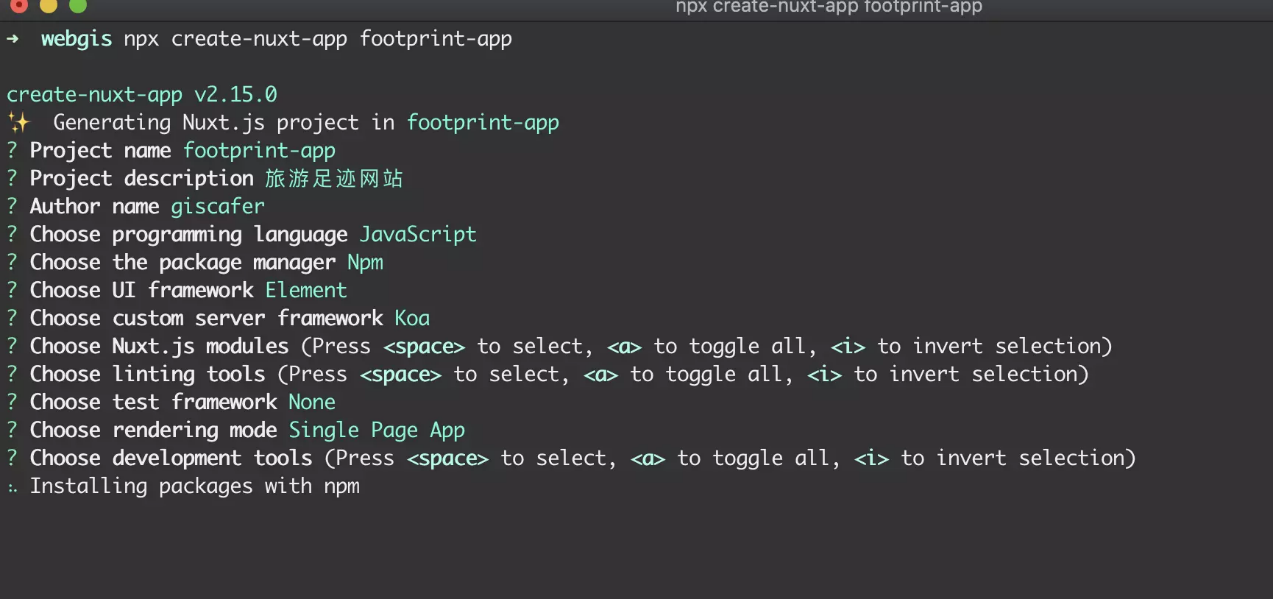
npm 版本大于5.2.0的用户可以直接执行：

1. npx **create**-nuxt-app footprint-app

yarn 的用户执行：

1. yarn **create** nuxt-app footprint-app

根据提示做技术选型，网站选型如图：



关键选择说明：

* 编程语言选择 JavaScript
* UI框架 选择饿了么的 Element，熟悉 antv 的可以选择 antv
* 服务端框架选择 Koa，熟悉 Express 的可以选择 Express
* 渲染方式选择 SPA（Single Page App），单页面应用，SSR的话需要点学习成本，为了降低难度所以这里介绍的是SPA的开发方式

通过上下箭头指定高亮选择，回车即可选择并进入下一步。选择完技术选项后会安装依赖，安装完依赖即创建完成工程了，然后根据提示启动工程。

**启动工程**

1. cd footprint-app
2. npm run dev



npm run dev



http://localhost:300

到此，工程基础框架创建完成。是不是很方便，这就是脚手架自动化的效率！重复的东西，做成自动化，只需要做一次就好了。

**工程规范与代码结构**

一些是创建好的过程的初始目录结构，每个目录结构的意思都注释了用途：

1. ├── README.md
2. ├── assets *// 此目录放置未编译的文件，如LESS、SASS 或 JavaScript。*
3. │ └── README.md
4. ├── components *// Vue.js 组件*
5. │ ├── Logo.vue
6. │ └── README.md
7. ├── layouts *// layouts 目录中的每个文件 (顶级) 都将创建一个可通过页面组件中的 layout 属性访问的自定义布局。*
8. │ ├── README.md
9. │ └── **default**.vue
10. ├── middleware *// 此目录包含应用程序中间件*
11. │ └── README.md
12. ├── nuxt.config.js *// nuxt 应用的配置文件*
13. ├── package.json *// 依赖*
14. ├── pages *// 页面代码文件*
15. │ ├── README.md
16. │ └── index.vue
17. ├── plugins *// 插件目录。用于组织那些需要在 根 vue.js应用 实例化之前需要运行的 Javascript 插件。(一般是引入第三方插件，比如组件库等）*
18. │ ├── README.md
19. │ └── element-ui.js
20. ├── server *// 后端服务*
21. │ └── index.js
22. ├── **static** *// 静态资源目录。不需要通过 Webpack 处理的静态资源文件，可以放置在 static 目录中*
23. │ ├── README.md
24. │ └── favicon.ico
25. └── store *// store 目录用于组织应用的 Vuex 状态树 文件*
26. └── README.md

Nuxt.js 目录结构很规范，基本定义好了每个目录的用途，开发过程需要编写什么功能性的代码，就在对应的一级目录下创建文件即可。每个一级目录下，也可以根据具体的代码结构设计进行管理。后续的章节在开发过程会说明怎么管理对应模块的代码。

如果还不够清晰，可以去官网详细了解一下[目录结构](https://zh.nuxtjs.org/guide/directory-structure)

**布局设计与开发**

了解了工程的目录结构之后，我们知道 layouts 目录是放页面布局功能的代码文件，接下来设计网站的页面布局并开发出雏形效果。

网站设计是管理后台的方式，通过 Element UI 提供的 [Comtainer](https://element.eleme.io/#/zh-CN/component/container) 布局组件，可以快速的实现以下的布局：



布局实现效果

**代码介绍**

编辑 ./layouts/default.vue 代码文件，将 template 中的代码替换成如下

1. <el-container style="height: 100%;">
2. <el-header style="text-align: right; font-size: 12px">
3. <el-dropdown>
4. <i class="el-icon-setting" style="margin-right: 15px"></i>
5. <el-dropdown-menu slot="dropdown">
6. <el-dropdown-item>查看</el-dropdown-item>
7. <el-dropdown-item>新增</el-dropdown-item>
8. <el-dropdown-item>删除</el-dropdown-item>
9. </el-dropdown-menu>
10. </el-dropdown>
11. <span>王小虎</span>
12. </el-header>
13. <el-container>
14. <el-aside width="200px" style="background-color: rgb(238, 241, 246)">
15. <el-menu :default-openeds="['1', '3']">
16. <el-submenu index="1">
17. <template slot="title">
18. <i class="el-icon-message"></i>导航一
19. </template>
20. <el-menu-item-group>
21. <template slot="title">分组一</template>
22. <el-menu-item index="1-1">选项1</el-menu-item>
23. <el-menu-item index="1-2">选项2</el-menu-item>
24. </el-menu-item-group>
25. <el-menu-item-group title="分组2">
26. <el-menu-item index="1-3">选项3</el-menu-item>
27. </el-menu-item-group>
28. <el-submenu index="1-4">
29. <template slot="title">选项4</template>
30. <el-menu-item index="1-4-1">选项4-1</el-menu-item>
31. </el-submenu>
32. </el-submenu>
33. <el-submenu index="2">
34. <template slot="title">
35. <i class="el-icon-menu"></i>导航二
36. </template>
37. <el-menu-item-group>
38. <template slot="title">分组一</template>
39. <el-menu-item index="2-1">选项1</el-menu-item>
40. <el-menu-item index="2-2">选项2</el-menu-item>
41. </el-menu-item-group>
42. <el-menu-item-group title="分组2">
43. <el-menu-item index="2-3">选项3</el-menu-item>
44. </el-menu-item-group>
45. <el-submenu index="2-4">
46. <template slot="title">选项4</template>
47. <el-menu-item index="2-4-1">选项4-1</el-menu-item>
48. </el-submenu>
49. </el-submenu>
50. <el-submenu index="3">
51. <template slot="title">
52. <i class="el-icon-setting"></i>导航三
53. </template>
54. <el-menu-item-group>
55. <template slot="title">分组一</template>
56. <el-menu-item index="3-1">选项1</el-menu-item>
57. <el-menu-item index="3-2">选项2</el-menu-item>
58. </el-menu-item-group>
59. <el-menu-item-group title="分组2">
60. <el-menu-item index="3-3">选项3</el-menu-item>
61. </el-menu-item-group>
62. <el-submenu index="3-4">
63. <template slot="title">选项4</template>
64. <el-menu-item index="3-4-1">选项4-1</el-menu-item>
65. </el-submenu>
66. </el-submenu>
67. </el-menu>
68. </el-aside>
69. <el-main>
70. <h3>布局演示</h3>
71. *<!-- nuxt 相当于路由，nuxt.js 会编译别的页面插入此处 -->*
72. <nuxt />
73. </el-main>
74. </el-container>
75. </el-container>

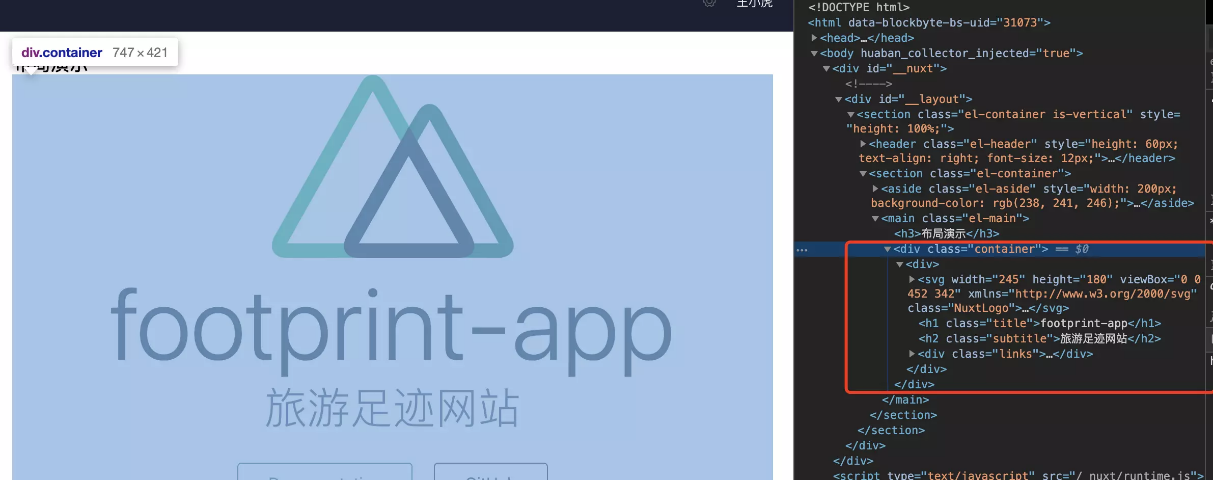
新增样式为：

1. .el-header {
2. background: #1b2034;
3. color: #fff;
4. line-height: 60px;
5. }
6. .el-aside {
7. color: #333;
8. }

由于nuxt.js布局文件有个小坑，渲染后的几个div高度不是容器html或body的100%高度，所以需要重写一下样式，还是放在 layouts/default.vue 的 style 下

1. **html**,
2. **body**,
3. **#\_\_nuxt**,
4. **#\_\_layout**,
5. **#\_\_layout** {
6. height: 100%;
7. margin: 0;
8. padding: 0;
9. }

到此就完成了布局页面的效果了。可以发现 pages/index.vue 的代码被编译放到了 <nuxt/> 下



nuxt

学习的同学可以通过布局代码提交的日记详情看本次代码的改动：[chore: 布局简单实现](https://github.com/gis520/footprint-app/commit/ebee308437d2f2fe731a2e4e032e52365328a8c2)

**小结**

好了，以上就是本篇的主要内容了。本篇学习主要是掌握 **Nuxt.js** 创建应用并立即起目录结构和开发规范，以及如果实现一个网站布局的模块开发。如果有哪些你希望深入了解的地方还未涉及到，希望你可以留言给我或者在微信群中讨论。