

爱旅行项目—前端打包部署

随着技术的不断变化，前端项目的工程结构设计和部署方式也在不断的变化中，为了适应企业开发的需求和提高开发效率，我们在爱旅行项目中引入了分布式部署，实现前后端环境的分离开发，这样不仅能提高我们的开发效率还能减少服务器之间的相互影响，同时还能保证数据安全等；但是随之而来的问题比较明显，前端项目怎样进行本地测试，加载类库？开发完成之后前端如何打包部署？这些问题将如何解决呢？

关于以上的问题，我们针对性的制定出解决方案，关于本地测试、环境部署，我们主要依赖 **npm**（包管理系统）、**webpack**（模块打包工具）来进行本地调试和文件的打包部署，接下来我们从以下几点进行讲解：本地项目初始化、本地环境运行、本地文件打包、文件部署；（注：**npm**、**webpack** 的下载、安装等说明请详见《**NPM 使用文档**》、《**webpack 使用文档**》）

1.1 本地项目初始化

当我们从 OSChina 上面拿到我们的项目工程之后（工程模板地址：<http://git.oschina.net/zhzbin/itripmould>）使用 **git clone** 将代码下载到本地环境 然后将 **itripmould** 文件夹在 VSCode 编译器中打开，并打开终端（如图 1）

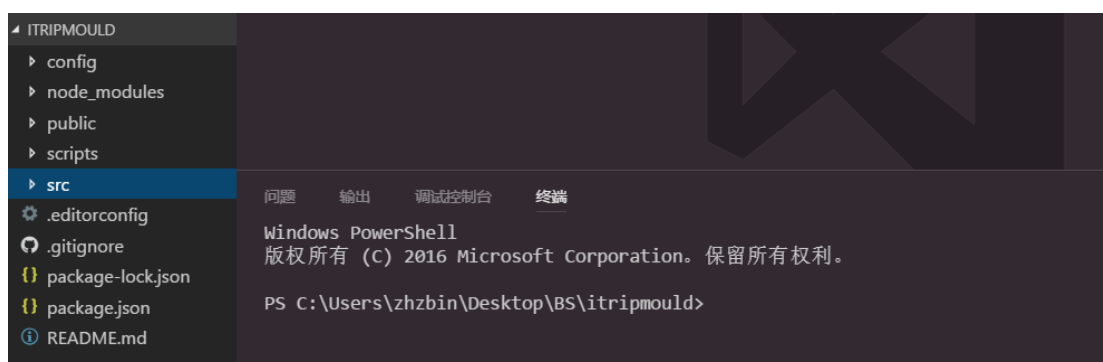


图 1

1. 在终端输入 **npm install** 之后，界面上会出现（如图 2）的界面效果，表明正在下载所需要的插件和需要的文件。

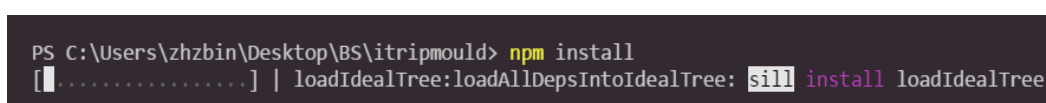


图 2

2. 插件安装完成之后左侧文件导航栏中会出现 `node_modules` 文件夹，此文件夹里面的文件就是我们初始化时加载到我们本地的插件库。

1.2 本地环境运行

完成项目的文件初始化之后，我们如何启动我们的环境实现本地调试呢？这里我们只需要在控制台中输入 **`npm start`** 即可启动项目，启动成功后会默认打开 (<http://localhost:3000>) 地址，同时，终端控制台会打印出部分日志信息（如图 3）

```
Warning in ./src/common/js/fetch.js
C:\Users\zhzbin\Desktop\BS\itripmould\src\common\js\fetch.js
  3:19  warning  'Button' is defined but never used  no-unused-vars
  51:36  warning  Invalid typeof comparison value    valid-typeof

X 2 problems (0 errors, 2 warnings)

Warning in ./src/pages/HotelList/ListItem.js
C:\Users\zhzbin\Desktop\BS\itripmould\src\pages\HotelList\ListItem.js
  42:15  warning  img elements must have an alt prop, either with meaningful text, or an empty string for decorative images  jsx-ally/img-has-alt
  83:15  warning  img elements must have an alt prop, either with meaningful text, or an empty string for decorative images  jsx-ally/img-has-alt

X 2 problems (0 errors, 2 warnings)

You may use special comments to disable some warnings.
Use // eslint-disable-next-line to ignore the next line.
Use /* eslint-disable */ to ignore all warnings in a file.
```

图 3

项目启动完成之后，我们可以在控制台按 **`Ctrl+C`** 停止服务，会提示 **终止批处理操作吗(Y/N)?** 输入 **`Y`** 即可停止服务；此时页面 <http://localhost:3000> 就不能再被访问；

1.3 本地文件打包

项目开发完成之后，需要将开发的文件进行打包部署到 `nginx`，如何进行打包部署呢？其实比较简单，主要使用 **`npm run build`** 即可进行文件的打包，打包依赖的配置文件是我们已经配置好了的图 4 是我们打包完成后的界面结构和打包过程中的 `log` 文件，其中左侧的文件导航栏中的 `build` 文件夹是我们打包后生成的文件目录，我们可以直接将 `build` 文件夹中的文件放到 `nginx` 服务器上，进行部署即可。同时我们也可以通过添加本地 `serve`（**不推荐**）进行打包文件测试具体信息如下：

`npm install -g serve` （插件安装）

`serve -s build` （打包运行）

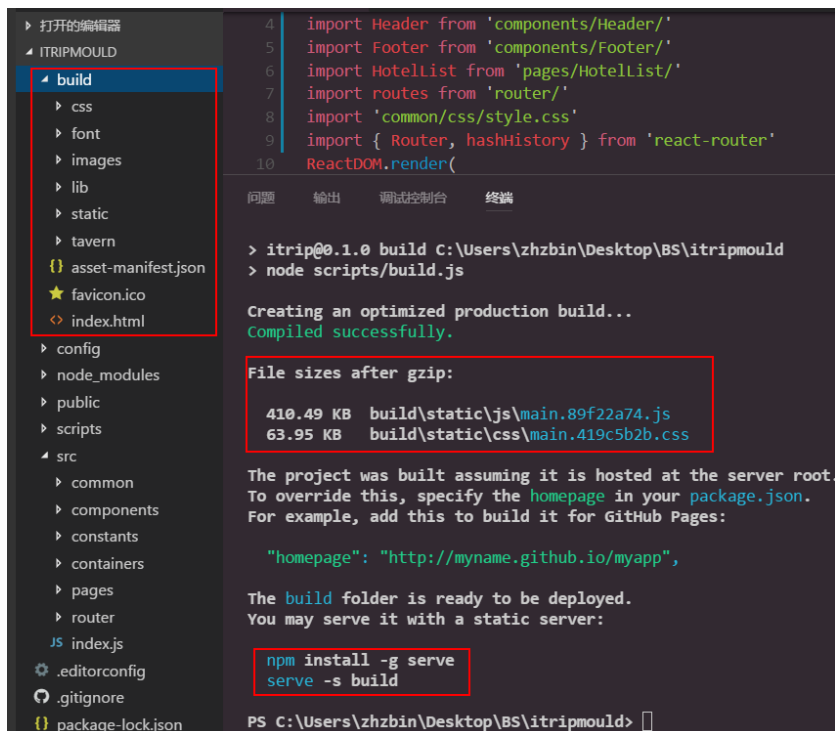


图 4

1.4 文件部署

将 build 文件夹下的文件拷贝到 Nginx 下，关于 Nginx 的详细配置请参照《Nginx 关键技术分析及运用》