

爱旅行项目一前端项目本地调试

Web 前端项目的开发使用分布式设计，使前端不能跟 server 端放到相同的域里，这样就会出现跨域访问的问题，那我们如何解决此类问题？怎样才能让前端页面直接访问后端的数据还不会有跨域的问题出现？这里我们使用代理来解决本地测试跨域访问的问题，如何配置代理？实施部署项目的时候我们使用哪种技术来代替本地代理？

关于本地代理我们这使用 http-proxy-middleware 插件来解决，实施部署时我们使用 Nginx 来实现本地代理访问，关于 Nginx 的配置主要参照《Nginx 关键技术分析及运用》文档，接下来我们就来讲一下本地代理的使用，在使用之前先初始化项目，这里简单介绍一下：

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）

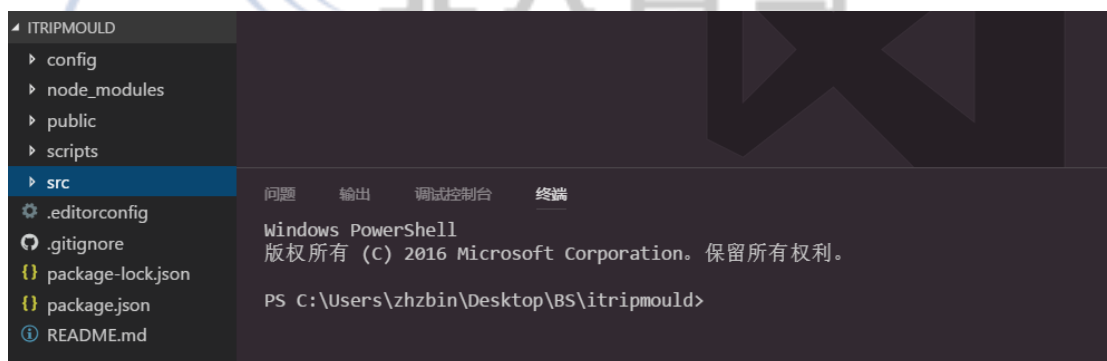


图 1

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。

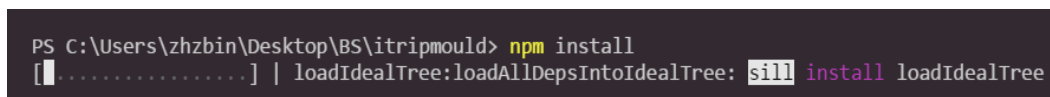
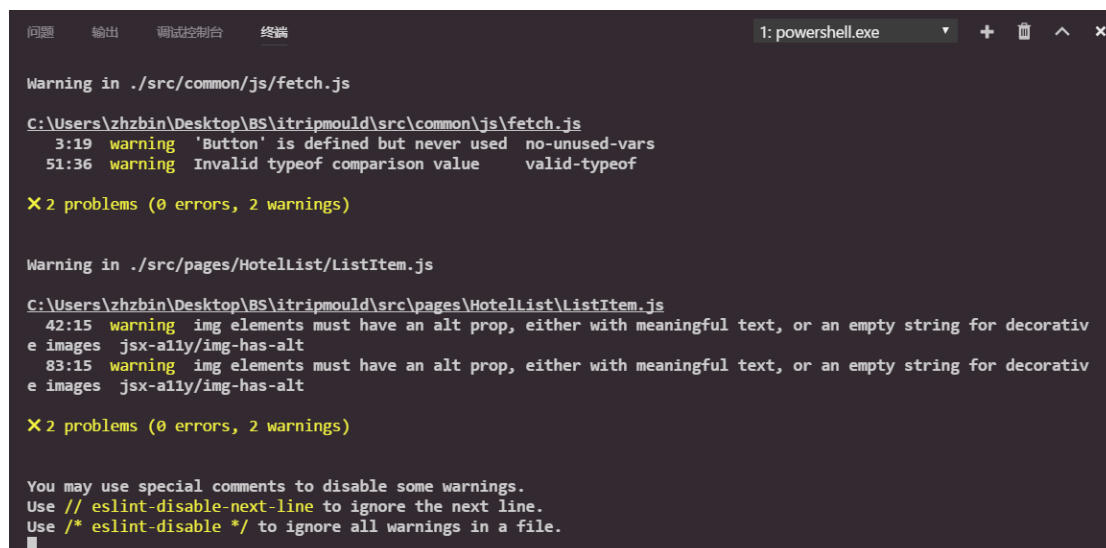


图 2

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？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)

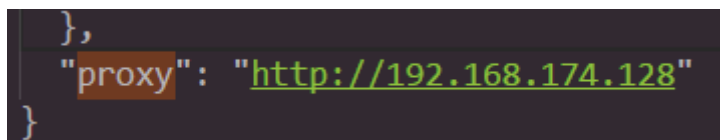
You may use special comments to disable some warnings.
Use // eslint-disable-next-line to ignore the next line.
Use /* eslint-disable */ to ignore all warnings in a file.
```

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 代理配置

代理服务器配置比较简单，只需要使用 `npm install --save-dev http-proxy-middleware`，我们的工程里面已经引入此文件，不需要大家去添加此插件，我们需要在 `package.json` 文件中去修改，在文件里面添加：`"proxy": "代理地址"`，如图 4



```
{
  "proxy": "http://192.168.174.128"
}
```

图 4

1.4 测试代理地址

代理地址配置完成之后，我们需要添加对应的测试方式，主要是测试代理地址是否配置成功，我们使用 `fetch` 函数进行数据请求，以 Rap 接口代理为例，正常我们发送 Rap 接口的地址为：<http://rapapi.org/mockjsdata/23181/api/hotel/queryhotelfeature>，我们 `package.json` 文件中代码里面的配置可以写成 `"proxy": http://rapapi.org/mockjsdata/23181/`，

然后我们使用 `npm start` 启动服务 `fetch` 请求编写为:

```
fetch("/api/hotel/queryhotelfeature").then(
  res=>{
    return res.json()
  }
).then(res=>{
  document.write(JSON.stringify(res))
})
```

页面显示信息为:

```
{"msg":"未登录请重新登录","errorCode":10000,"data|1-10":[{"id":22247,"name":"胖成球"},{"id":22247,"name":"undefined"}],"string":"★★★★★★★★","success":true}
```

跟我们直接访问此请求数据是一致的, 说明我们已经将代理配置成功了;

