npm 国内镜像切换|nvm 使用笔记

slongzhang\_

于 2022-03-06 14:41:23 发布

770

收藏 1

文章标签： javascript chrome 前端

版权

临时使用

在 npm install XXX 时加入–registry URL即可，不会影响到本地配置，如：

npm --registry https://registry.npm.taobao.org install express

1

2. 持久使用

永久修改本地npm的配置：

npm config set registry https://registry.npmmirror.com/

1

查看npm源

npm config get registry

1

重置为官方源

npm config set registry https://registry.npmjs.org/

1

其他源镜像

# 腾讯镜像

npm config set registry http://mirrors.cloud.tencent.com/npm/

# 淘宝镜像 (官网 https://npmmirror.com/)

npm config set registry https://registry.npmmirror.com/

# 华为镜像

npm config set registry https://mirrors.huaweicloud.com/repository/npm/

NVM下载安装

下载链接 https://github.com/coreybutler/nvm-windows/releases

可下载以下版本

nvm-noinstall.zip：绿色免安装版，但使用时需要进行配置。(配置方法自行百度)

nvm-setup.zip：安装版，推荐使用

使用nvm

nvm list 命令 - 显示版本列表

nvm list // 显示已安装的版本（同 nvm list installed）

nvm list installed // 显示已安装的版本

nvm list available // 显示所有可以下载的版本

1

2

3

nvm install 命令 - 安装指定版本nodejs

nvm install 14.5.0 // 安装14.5.0版本node

nvm install latest // 安装最新版本node

1

2

nvm use 命令 - 使用指定版本node

nvm use 14.5.0 // 使用14.5.0版本node

1

nvm uninstall 命令 - 卸载指定版本 node

nvm uninstall 14.5.0 // 卸载14.5.0版本node

————————————————

版权声明：本文为CSDN博主「slongzhang\_」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_35606400/article/details/123309885

### 安装 `npm`

**`npm` 全称为 `Node Package Manager`，是一个基于`Node.js`的包管理器，也是整个`Node.js`社区最流行、支持的第三方模块最多的包管理器。**

```

npm -v

```

### 由于网络原因 安装 `cnpm`

```

npm install -g cnpm --registry=https://registry.npm.taobao.org

```

### 安装 `vue-cli`

```

cnpm install -g @vue/cli

```

### 安装 `webpack`

**`webpack` 是 `JavaScript` 打包器(module bundler)**

```

cnpm install -g webpack

```

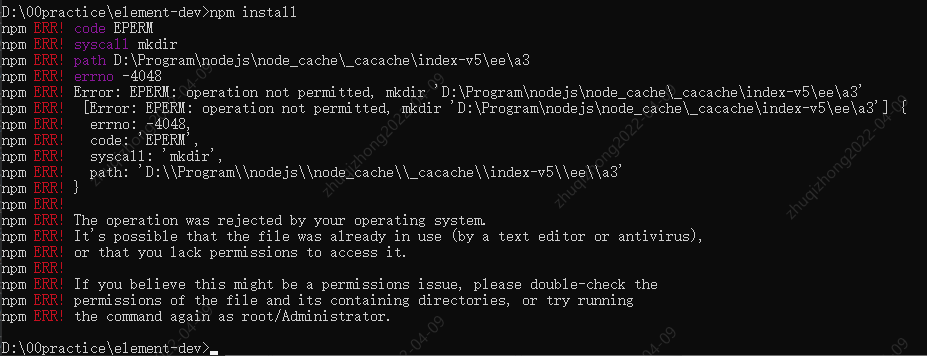
D:\00practice\element-dev>tskill 6296

D:\00practice\element-dev>npm config set registry https://registry.npm.taobao.org

D:\00practice\element-dev>npm config get registry

https://registry.npm.taobao.org/

D:\00practice\element-dev>npm install



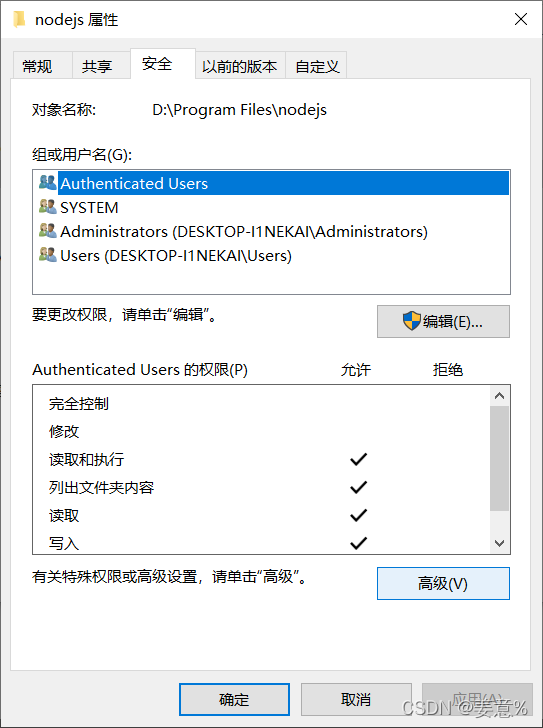
npm install为什么要管理员权限

# **npm安装包时需要管理员权限解决办法**

在使用npm下载包时老是出错



每次安装都要用管理员身份运行，**主要因为安装nodejs时没有授予写的权限，找到nodejs文件夹更改权限即可**



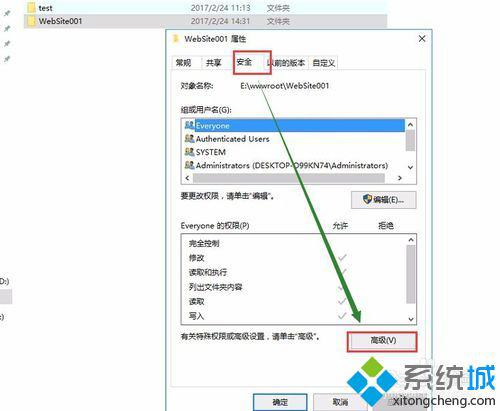
# **win10提示“无法枚举容器中的对象 访问被拒绝”如何解决**

发布时间：2017-02-27 18:11发布者：系统城-小群浏览数：114731

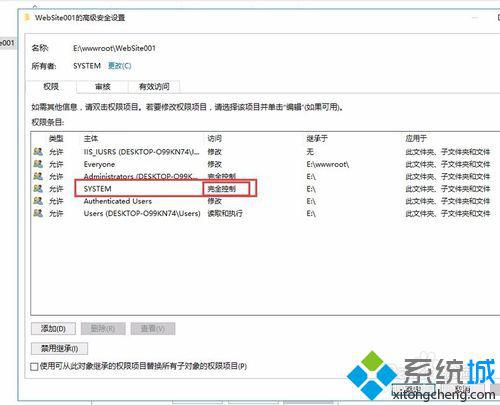
不少用户在[windows10系统](http://www.xitongcheng.com/win10/" \o "" \t "http://www.xitongcheng.com/jiaocheng/_blank)电脑中添加用户权限时，遇到了“无法枚举容器中的对象 访问被拒绝”这个错误。那么，我们该如何操作呢？接下来，就随小编看看win10提示“无法枚举容器中的对象 访问被拒绝”问题的解决方法吧！

**具体如下:**

1、右击文件，选择“属性”--“安全”--“高级”；



2、查看哪个用户的权限是“完全控制”，我这里是“system”；



3、选择“更改----“高级”----“立即查找”，选择管理员账户“system”；





4、勾选“替换子容器和对象的所有者”，点击“确认”，之后会弹出来一个对话框，点击“是”。



  win10提示“无法枚举容器中的对象 访问被拒绝”的解决方法就为大家介绍到这里了。遇到类似问题的朋友们，不必太过紧张，只要参考下小编的教程就可以了。

**vue-cli创建的项目npm run dev运行很慢**

千次阅读

2020-03-03 15:23:20

**vue-cli创建的项目npm run dev刚开始运行正常，之后突然运行要花很长时间。**后来发现是cpmn的原因。有次npm i装包的时候失败了，于是就用了cnpm i来装包，装包后再跑项目就很慢了。

解决方法：删除node\_modules文件，npm i重新安装包

# **Vue项目build执行编译时卡住，长时间不动**

IMG_256

每次npm run dev 启动项目的时候就会building modules，而且会很慢，不知道是什么原因

npm

* 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
* 允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
* 允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用

npm命令

* npm -v 来测试是否成功安装
* 查看当前目录已安装插件：npm list
* 更新全部插件： npm update [ --save-dev ]
* 使用 npm 更新对应插件： npm update <name> [ -g ] [ --save-dev]
* 使用 npm 卸载插件： npm uninstall <name> [ -g ] [ --save-dev ]

cnpm

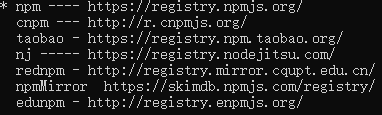
* 淘宝团队做的国内镜像，因为npm的服务器位于国外可能会影响安装。淘宝镜像与官方同步频率目前为 10分钟 一次以保证尽量与官方服务同步。
* 安装：命令提示符执行
* npm install cnpm -g --registry=https://registry.npm.taobao.org
* cnpm -v 来测试是否成功安装

通过改变地址来使用淘宝镜像

* npm的默认地址是https://registry.npmjs.org/
* 可以使用npm config get registry查看npm的仓库地址
* 可以使用npm config set registry https://registry.npm.taobao.org来改变默认下载地址，达到可以不安装cnpm就能采用淘宝镜像的目的，然后使用上面的get命令查看是否成功。

nrm

* nrm包安装命令： npm i nrm -g
* nrm能够管理所用可用的镜像源地址以及当前所使用的镜像源地址，但是只是单纯的提供了几个url并能够让我们在这几个地址之间方便切换
* nrm ls即nrm list，查看所有可用的镜像，并可以切换。\*号表示当前npm使用的地址，可以使用命令nrm use taobao或 nrm use npm来进行两者之间的切换。



<p>nrm ls命令</p>

</li>

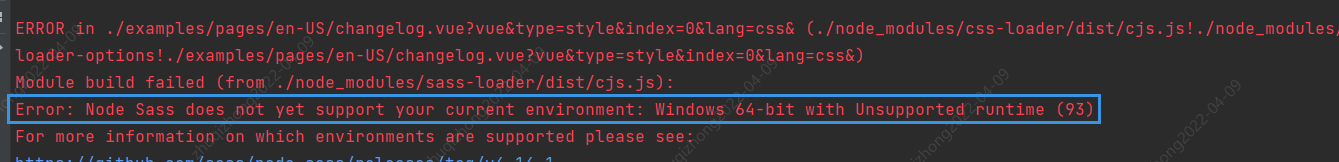
* + 1
  + 2

-g -S -D

* -g：全局安装。 将会安装在C：\ Users \ Administrator \ AppData \ Roaming \ npm，****并且写入系统环境变量****；非全局安装：将会安装在当前定位目录;全局安装可以通过命令行任何地方调用它，本地安装将安装在定位目录的node\_modules文件夹下，通过要求调用;
* -S：即npm install module\_name --save,写入package.json的dependencies ,dependencies 是需要发布到生产环境的，比如jq，vue全家桶，ele-ui等ui框架这些项目运行时必须使用到的插件就需要放到dependencies
* -D：即npm install module\_name --save-dev,写入package.json的devDependencies ,devDependencies 里面的插件只用于开发环境，不用于生产环境。比如一些babel编译功能的插件、webpack打包插件就是开发时候的需要，真正程序打包跑起来并不需要的一些插件。

为什么要保存在package.json 因为node\_module包实在是太大了。用一个配置文件保存，只打包安装对应配置文件的插件，按需导入。

Error: Node Sass does not yet support your current environment: Windows 64-bit with Unsupported runtime (93)



PS D:\00practice\element-dev> npm uninstall --save node-sass

up to date in 4s

PS D:\00practice\element-dev> **npm install --save node-sass**

npm ERR! code EPERM

npm ERR! syscall open

npm ERR! path D:\Program\nodejs\node\_cache\\_cacache\index-v5\4e\44\9b28629a805b5c2f01d69ea8c434bb87aa5ec0c1212d6b0725c092993858

npm ERR! errno -4048

npm ERR! Error: EPERM: operation not permitted, open 'D:\Program\nodejs\node\_cache\\_cacache\index-v5\4e\44\9b28629a805b5c2f01d69ea8c434bb87aa5ec0c1212d6b0725c092993858'

npm ERR! [Error: EPERM: operation not permitted, open 'D:\Program\nodejs\node\_cache\\_cacache\index-v5\4e\44\9b28629a805b5c2f01d69ea8c434bb87aa5ec0c1212d6b0725c092993858'] {

npm ERR! errno: -4048,

npm ERR! code: 'EPERM',

npm ERR! syscall: 'open',

npm ERR! path: 'D:\\Program\\nodejs\\node\_cache\\\_cacache\\index-v5\\4e\\44\\9b28629a805b5c2f01d69ea8c434bb87aa5ec0c1212d6b0725c092993858',

npm ERR! requiredBy: '.'

npm ERR! }

npm ERR!

npm ERR! The operation was rejected by your operating system.

npm ERR! It's possible that the file was already in use (by a text editor or antivirus),

npm ERR! or that you lack permissions to access it.

npm ERR!

npm ERR! If you believe this might be a permissions issue, please double-check the

npm ERR! permissions of the file and its containing directories, or try running

**npm ERR! the command again as root/Administrator.**

**端口为什么会被占用**

**为什么**

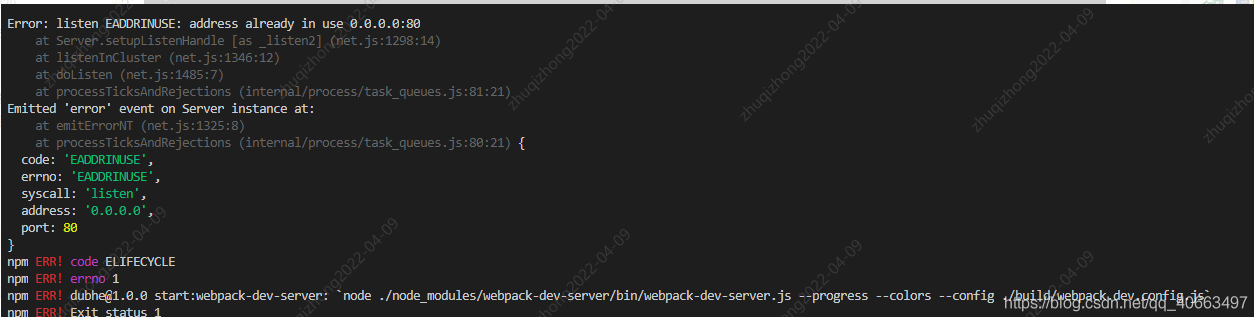
**Error: listen EADDRINUSE: address already in use**

百慕达奶油

于 2019-12-12 09:49:17 发布

收藏 1

经常遇到这个报错，在此记录一下



根据错误提示，Error: listen EADDRINUSE: address already in use 0.0.0.0:80，意思是80端口已经被占用。这时，我寻找的解决方案为：

1、win+r 输入cmd 打开终端

2、运行netstat -ano，找到报错信息提示的端口号那一行，记住最后那个数字

　3.接下来运行tskill “最后那个数字”，所以这里我运行的是tskill 22132

　回到IDE中重启服务就可以啦



————————————————

版权声明：本文为CSDN博主「百慕达奶油」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_40663497/article/details/103503405>

**解决端口号被占用的方法：listen EADDRINUSE: address already in use :::5000**

netstat -ano|findstr "5000"

tskill pid

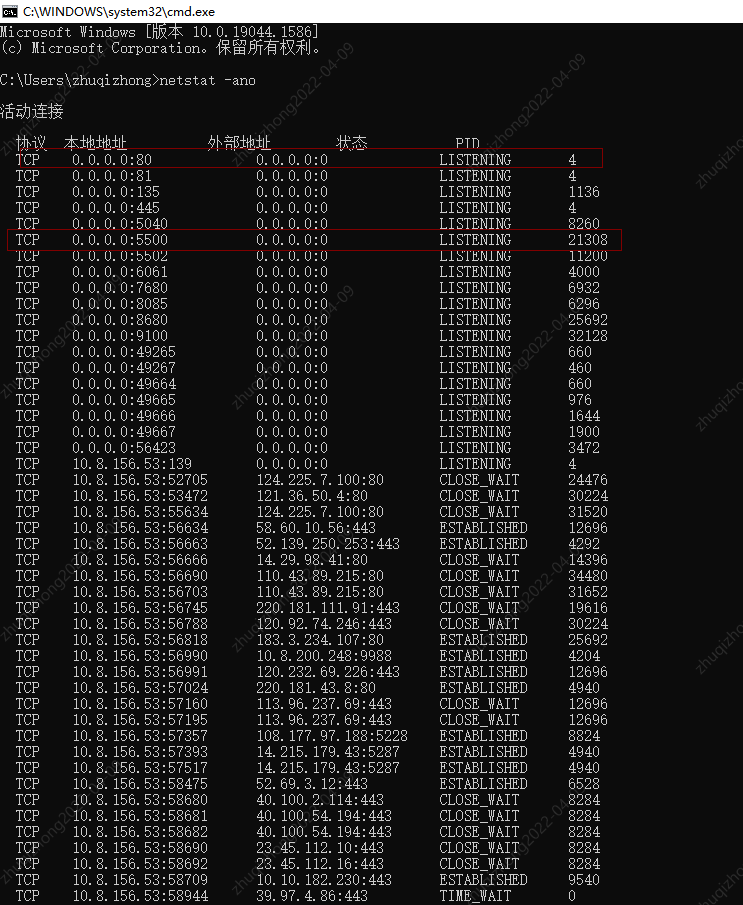


————————————————

版权声明：本文为CSDN博主「Damao97」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_41258179/article/details/96761871>





[报错]执行npm install.报If you are behind a proxy,please make sure that the proxy cinfig is set property.

肆〇

于 2021-08-23 14:30:32 发布

1956

收藏 6

文章标签： redis 数据库 缓存

版权

问题：运行npm 命令报错。运行npm cache clean --force清理缓存后，仍然报错。

原因一：代理的默认地址为proxy=http://127.0.0.1:8087 。

解决方法：运行$ npm config set proxy null 代理置为空。

原因二：网络不行，网络质量差或者公司的网有设置网络权限

解决方法：让别人把已经下好的node\_moudles打包发给你，直接覆盖到相应的项目文件位置中。

————————————————

版权声明：本文为CSDN博主「肆〇」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/m0\_51660523/article/details/119863234

If you are behind a proxy, please make sure that the

npm install webpack报错if you are behind a proxy,please make sure that the proxy cinfig is setproperly

Tempta36

于 2020-03-05 15:48:19 发布

11341

收藏 8

分类专栏： node

版权

node

专栏收录该内容

1 篇文章0 订阅

订阅专栏

npm install webpack -g报错if you are behind a proxy,please make sure that the 'proxy' cinfig is set properly.

错误信息提示：若配置了代理，请确认代理被合理配置。因此去设置npm代理

执行npm config set proxy http://proxy.example.com:8080

参考https://qiita.com/tenten0213/items/7ca15ce8b54acc3b5719

然后重新执行npm install webpack -g，又报错

提示：服务器出错，重新配置npm镜像地址

执行npm config set registry http://registry.npmjs.org/

终于，webpack安装成功了。



————————————————

版权声明：本文为CSDN博主「Tempta36」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/Tempta36/article/details/104676312>

Fix the upstream dependency conflict, or retry this command with --force, or --legacy-peer-deps

在NPM v7中，现在默认安装peerDependencies。

在很多情况下，这会导致版本冲突，从而中断安装过程。

--legacy-peer-deps标志是在v7中引入的，目的是绕过peerDependency自动安装；它告诉 NPM 忽略项目中引入的各个modules之间的相同modules但不同版本的问题并继续安装，保证各个引入的依赖之间对自身所使用的不同版本modules共存。

————————————————

版权声明：本文为CSDN博主「longfei815」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/jolinoy/article/details/123951491>

ERESOLVE与npm@7有关的问题很常见，因为npm7.x对某些事情比npm6.x更严格。

1.--legacy-peer-deps标志传递给npm， npm i --legacy-peer-deps。

2.使用npx指定npm的版本。例如：npx -p npm@6 npm i --legacy-peer-deps。

————————————————

版权声明：本文为CSDN博主「web13985085406」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/web13985085406/article/details/123350608>

在使用npm install时遇到的问题

npm ERR! code ERESOLVE

npm ERR! ERESOLVE unable to resolve dependency tree

npm ERR!

npm ERR! While resolving: react\_staging@0.1.0

npm ERR! Found: react-scripts@1.1.5

roject

npm ERR!

npm ERR! Fix the upstream dependency conflict, or retry

npm ERR! this command with --force, or --legacy-peer-deps

npm ERR! to accept an incorrect (and potentially broken) dependency resolution.

找了好久才解决，原来因为npm7.x对某些事情比npm6.x更严格。

通常，最简单的解决方法是将--legacy-peer-deps标志传递给npm(e.g.，npm i --legacy-peer-deps），或者使用npm@6。

提示：使用npm@6不需要卸载npm@7。使用npx指定npm的版本。例如：npx -p npm@6 npm i --legacy-peer-deps。

如果这不能立即起作用，也许可以先删除node\_modules和package-lock.json。它们将被重新创建。

# **npm install:Could not resolve dependency：peer... 原因和解决方案**

[滴答滴答就是我 IMG_257](https://juejin.cn/user/2242659451959934" \t "https://juejin.cn/post/_blank)

2022年01月02日 17:20 ·  阅读 384

关注

新down公司一个之前的项目，在公司的电脑上npm install是没问题的，到家里的电脑npm install 就报错； 报错日志如下：

npm ERR! code ERESOLVE npm ERR! ERESOLVE unable to resolve dependency tree npm ERR! npm ERR! While resolving: shopping-mall@1.0.0 npm ERR! Found: echarts@5.2.2 npm ERR! node\_modules/echarts npm ERR! echarts@"^5.2.1" from the root project npm ERR! npm ERR! Could not resolve dependency: npm ERR! peer echarts@"^4.1.0" from vue-echarts@4.1.0 npm ERR! node\_modules/vue-echarts npm ERR! vue-echarts@"^4.0.1" from the root project npm ERR! npm ERR! Fix the upstream dependency conflict, or retry npm ERR! this command with --force, or --legacy-peer-deps npm ERR! to accept an incorrect (and potentially broken) dependency resolution.

家里的node 和npm 跟公司不一样 上述报错日志中有个关键字眼：**this command with --force, or --legacy-peer-deps**

于是我查了一下是什么意思： 搜索得出这样的答案：

****修复上游依赖性冲突，或使用--force或--legacy-peer-deps重试此命令，以接受不正确的（并且可能会损坏的）依赖性解析。****

-f或–force参数将强制npm获取远程资源，即使磁盘上存在本地副本也是如此。

同时，用于的文档--legacy-peer-deps说：

–legacy-peer-deps：安装时忽略所有peerDependencies，其样式为npm版本4到版本6。

在新版本的npm（v7）中，默认情况下，npm install当遇到冲突的peerDependencies时将失败。以前不是那样的。

两者之间的区别如下-

--legacy-peer-deps：安装时忽略所有peerDependencies，采用npm版本4到版本6的样式。 --strict-peer-deps：遇到任何冲突的peerDependencies时，失败并中止安装过程。默认情况下，npm仅会因根项目的直接依赖关系而导致的peerDependencies冲突而崩溃。

以上是英文文献翻译过来的，是有点生硬：我简单总结一下： 在新版本的npm（v7）中，默认情况下，npm install当遇到冲突的peerDependencies时将失败。不会继续安装，并提示：

Fix the upstream dependency conflict, or retry

this command with --force, or --legacy-peer-deps

to accept an incorrect (and potentially broken) dependency resolution.复制代码

****这样的关键字，这是npm版本的依赖冲突的提示使然，****

****那么npm：何时使用--force和--legacy-peer-deps？****

****--force 会无视冲突，并强制获取远端npm库资源，即使本地有资源也会覆盖掉****

****--legacy-peer-deps：安装时忽略所有peerDependencies，忽视依赖冲突，采用npm版本4到版本6的样式去安装依赖，已有的依赖不会覆盖。****

****建议用--legacy-peer-deps 比较保险一点，反正我用了就安装成功了；****

初学者如果是通过网上教程来学习webpack，第一次用webpack打包时通常会遇到下面这样的问题：

实际上出错信息已经说明了问题原因：

Invalid configuration object. Webpack has been initialised using a configuration object that does not match the API schema.

这一段的意思是webpack.config.js错误，原因是这个配置文件的版本和我们当前安装的webpack的版本不匹配。

configuration.module has an unknown property ‘loaders’.

接下来这段我们只需要看前面一句，意思是webpack.config.js这个配置文件里的module属性有一个未知的配置项loaders，原因嘛，就是我们当前安装的webpack版本已经去掉了这个配置。

如https://www.runoob.com/w3cnote/webpack-tutorial.html这里的教程，在开始已经声明了所用的webpack版本：

检查一下我们工程里webpack的版本：

{

"name": "angular-webpack",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"dependencies": {

"angular": "^1.6.10",

"css-loader": "^0.28.11",

"style-loader": "^0.21.0",

"webpack": "^4.6.0"

}

}

版本是4.6.0，看来这些webpack的配置并没有做成向前兼容，下面给出两个解决方案：

一个方法是去webpack的官网看最新的配置，比如上面的问题可以在这里看到（官网的配置都没说支持loaders，硬要套3.0版本的写法这不是自寻死路？不过还是要吐槽一下为什么不能向前兼容呢）。

另一个方法就是去下载指定版本的webpack：npm如何下载指定版本插件。

当然，有时间的话页可以自己慢慢调试，比如自己在webpack.config.js里手动去掉出错信息提示的loaders，不过效率就呵呵了。

————————————————

版权声明：本文为CSDN博主「讨厌走开啦」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/lqlqlq007/article/details/80091899