●星河

1、切换到星河地址

npm set registry http://180.213.8.245:65515

星河私服172地址

npm set registry http://172.16.66.33:4873

2、安装星河组件

npm i gw-uicomponent -S //1.0

npm i galaxy-ui -S //2.0

3、切换回淘宝镜像

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

4、再本地安装

npm install

●npm install不成功的有可能原因

npm install node-sass@4.14.1 --save-dev

cnpm install

启动项目即可

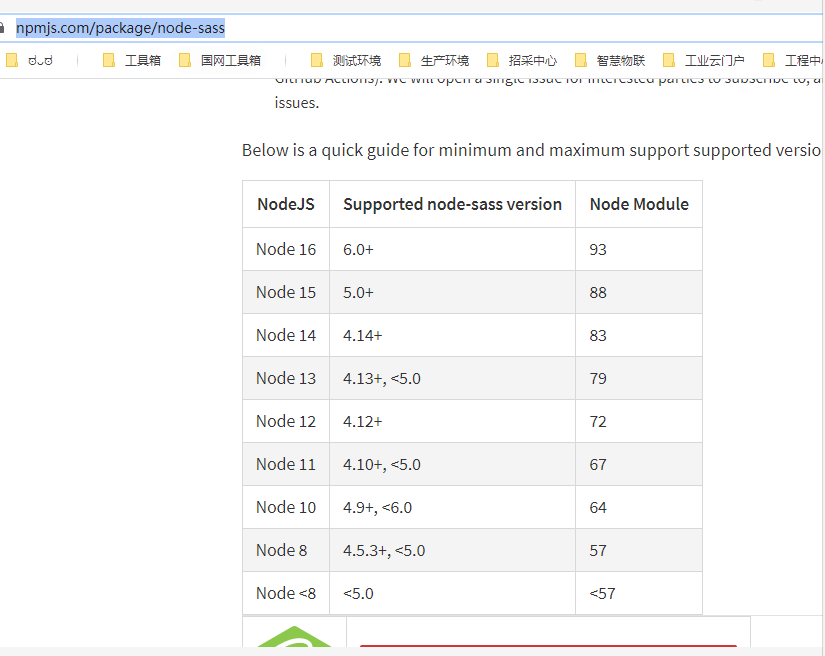
注：安装的node-sass版本要和本地node版本一致

可先查看node -v

例如node版本为v14.16.1

[则安装node-sass@4.14.1](mailto:则安装node-sass@4.14.1)

版本对照表https://www.npmjs.com/package/node-sass



**如果安装node sass不成功，先运行以下命令，全局安装windows构建工具**

**cnpm install -global -production windows-build-tools**

**。在命令行工具中运行npm install –global –production windows-build-tools （全局安装windows构建工具）。**

●查看npm镜像源

npm get registry

●清理npm缓存

npm cache clean -f

●查看npm安装路径

可通过命令 npm config get prefix 查看 npm 安装路径（npm config set prefix 可设置 npm 安装路径）；

2. 打开该路径，在 npm 目录下有个 node\_modules 文件夹，全局安装的所有模块都在该文件夹中 ;

●查看版本号

npm -v

node -v

●直接克隆一个分支，这样clone完，修改然后提交会直接提交到这个分支里

git clone -b 分支名 路径

Dklskdfj

● 开发环境devDependencies和生产环境dependencies

--save 等于 -S

--save-dev 等于 -D

一、模式

运行webpack命令时，一定要指定模式。

webpack --mode development  
webpack --mode production  
二、--save -dev

--save：将保存配置信息到pacjage.json。默认为dependencies节点中。

--dev：将保存配置信息devDependencies节点中。

因此：

--save：将保存配置信息到pacjage.json的dependencies节点中。

--save-dev：将保存配置信息到pacjage.json的devDependencies节点中。

dependencies：运行时的依赖，发布后，即生产环境下还需要用的模块

devDependencies：开发时的依赖。里面的模块是开发时用的，发布时用不到它。

<https://www.cnblogs.com/jpwz/p/12411816.html>

使用任何框架之前，都要经历环境搭建的过程，而在前端框架的环境搭建中（比如React、Vue等），NPM是必不可少的依赖包管理工具。我们在使用NPM命令安装依赖包时，经常会用到 -S 或者是 -D 这些指令，例如，我们安装webpack时会执行以下命令：

npm install -D webpack

对于前端初学者来说，可能还不太理解什么时候使用 -D ，什么时候使用 -S ，以及二者之间的区别。这就和我们这篇文章要讲的主题有关系了，NPM中的 dependencies 和 devDependencies 配置。

## 1. 什么是NPM

简单来说，NPM（Node Package Manager）是包含在Node.js里面的一个包管理工具，NPM会随着Node.js一起安装。NPM为开发者提供了一个代码模块共享的大平台，当我们项目中需要使用某个模块（JavaScript包）时，可以直接使用NPM包管理工具来下载对应的包并安装，我们也可以把自己用Node.js写的代码发布到平台上供他人使用。

## 2. package.json文件

搭建一个前端项目之前，通常会在项目的根目录下生成一个名为package.json的文件作为NPM包的描述文件，使用该文件来定义项目信息、配置包依赖关系。package.json文件可以自己手动创建，也可以使用命令来创建：

npm init

文件中包含了NPM包的基本信息（项目名称、版本号、项目描述、作者）和依赖管理，例如：

{

"name": "demo",

"version": "1.0.0",

"devDependencies": {

"webpack": "^4.29.6"

}

}

在package.json文件中，所有的依赖包都会在 dependencies 和 devDependencies 的配置项中进行管理，它们的意思是：

* **dependencies：** 表示生产环境下的依赖管理；
* **devDependencies：** 表示开发环境下的依赖管理；

## 3. 开发环境和生产环境

很多同学不太理解什么是开发环境和生产环境，简单来说，就是在项目的开发阶段就是开发环境；项目上线了，开始正式提供对外服务，上线后的阶段就是生产环境。在生产环境下，一般会关掉错误报告，打开错误日志等操作。

devDependencies**配置的是开发环境**，安装项目开发时所依赖的模块。比如像webpack工具，只是用来构建项目和打包，这些都是在开发阶段才使用的，等项目上线后就用不到webpack工具了，那么我们就可以把webpack安装到开发环境中，使用 --save-dev 命令安装到devdependencies下，命令语法：

npm install --save-dev packageName

# 简写

npm i -D packageName

dependencies**配置的是生产环境**，安装项目运行时所依赖的模块。比如jQuery库，等项目上线以后依然是要继续使用的，我们就要安装在生产环境中，如果没有把需要的依赖安装到生产环境中，项目上线运行时就有可能会报错。使用 --save 命令安装到 dependencies 下，命令语法：

npm install --save packageName

# 简写

npm i -S packageName

注：上面的 -S ，是大写的S

## 总结

| **配置项** | **命令** | **描述** |
| --- | --- | --- |
| devDependencies | --save-dev 简写 -D | 开发环境，管理的依赖包仅在开发阶段有效 |
| dependencies | --save 简写 -S | 生产环境，管理的依赖包在项目上线后依然有效 |
| 掌握了原理，在开发中我们就可以根据需求，来正确使用指令安装依赖包，防止以后出现不可预知的问题。 |  |  |

●npm 安装指定版本

使用@加版本号

npminstall--save[esri-loader@1.0.0](mailto:esri-loader@1.0.0)

●卸载NPM包

npm uninstall css-loader --save-dev

这样会卸载依赖包中的css-loader包，并自动将package.json中的依赖关系去掉

█

●vuecli官网

<https://cli.vuejs.org/>

切换远程分支

https://www.cnblogs.com/whm-blog/p/11641818.html

git checkout -b 要切换的分支名 origin/要切换的分支名

█git 修改密码后

管理员身份打开cmd

git config --system --unset credential.helper

git config --global credential.helper store

这是清空以前的登陆信息

这样，再执行git操作时,例如git pull，系统会提示输入用户名和密码

之后就能正常操作了

█git设置代理

<https://blog.csdn.net/weixin_45685193/article/details/120606369>

**取消代理**  
当你哪天不想用代理了，可以输入以下两条指令取消http代理

git config --global --unset http.proxy

git config --global --unset https.proxy

滋滋滋