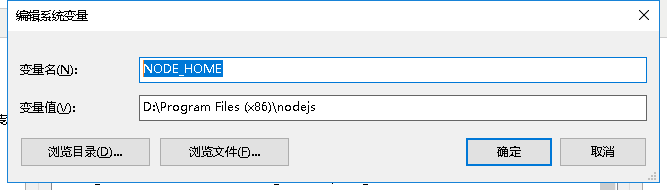
## 一、搭建node.js开发环境

windows系统，去网站https://nodejs.org/en/download/，下载对应的安装程序，并安装Windows Installer (.msi)

接下来，开始配置环境变量。

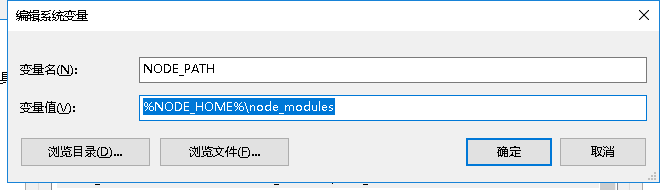
1. 配置“NODE\_HOME”为：“D:\Program Files (x86)\nodejs”



我这里，node.js安装在“D:\Program Files (x86)\nodejs”目录下，这根据你的具体安装路径而设置。

2.配置“NODE\_PATH”为： %NODE\_HOME%\node\_modules

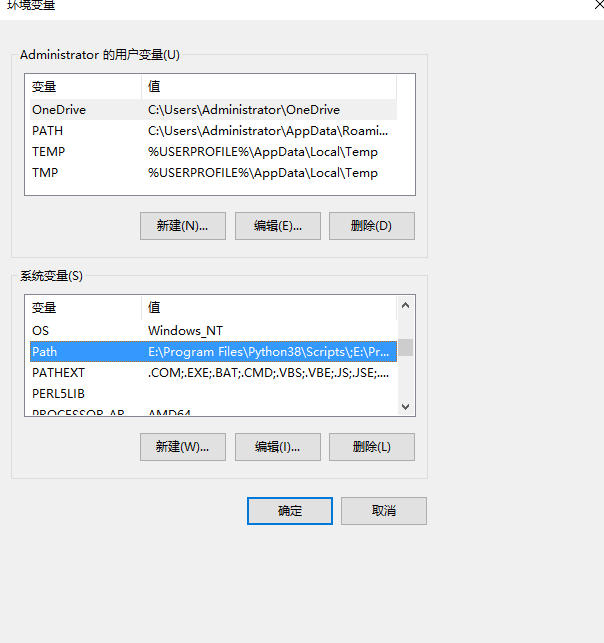
这关系到使用npm的命令，如果配置错了，安装的node依赖的目录就不正确。

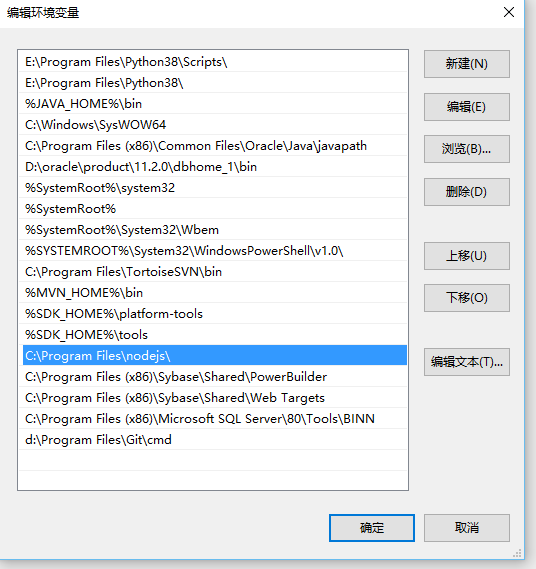


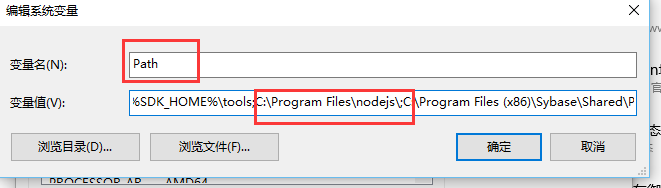
3、配置Path，追加 ;%NODE\_HOME%\;%NODE\_PATH%\; 这两项

或者

直接在path配置







4.检测环境

在CMD中输入 node -v 和 npm -v

如果出现版本号，则说明安装成功



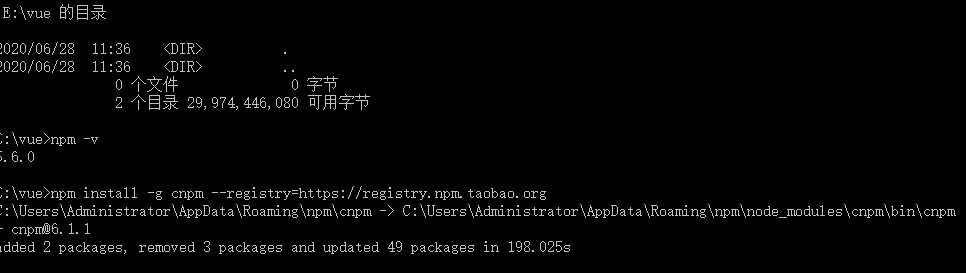
## 二、搭建Vue环境

### 2.1 安装cnpm淘宝镜像

参照https://npm.taobao.org/

在CMD中运行

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



### 2.2 安装Vue脚手架

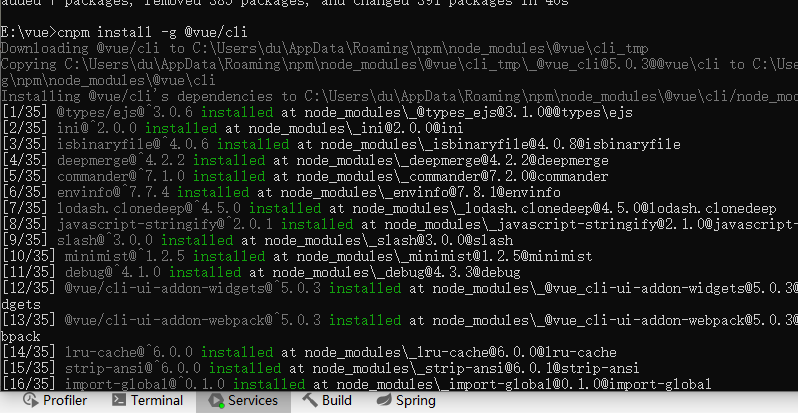
全局安装过旧版本的 vue-cli(1.x 或 2.x)要先卸载它，否则跳过此步：

查看版本vue -V

npm uninstall vue-cli –g

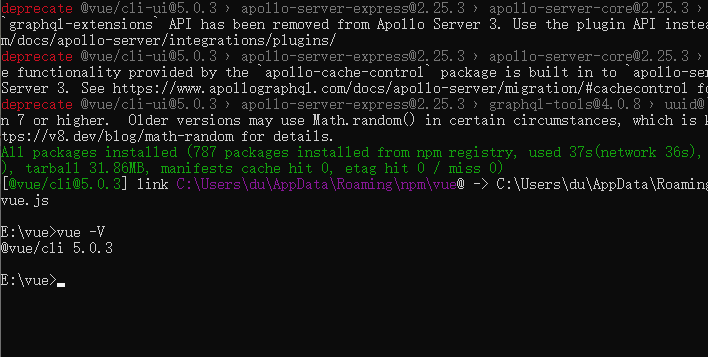
安装vue-cli 有两种，分别是vue-cli2.0和vue-cli3.0，任选一种，目前我们用vue-cli3.0

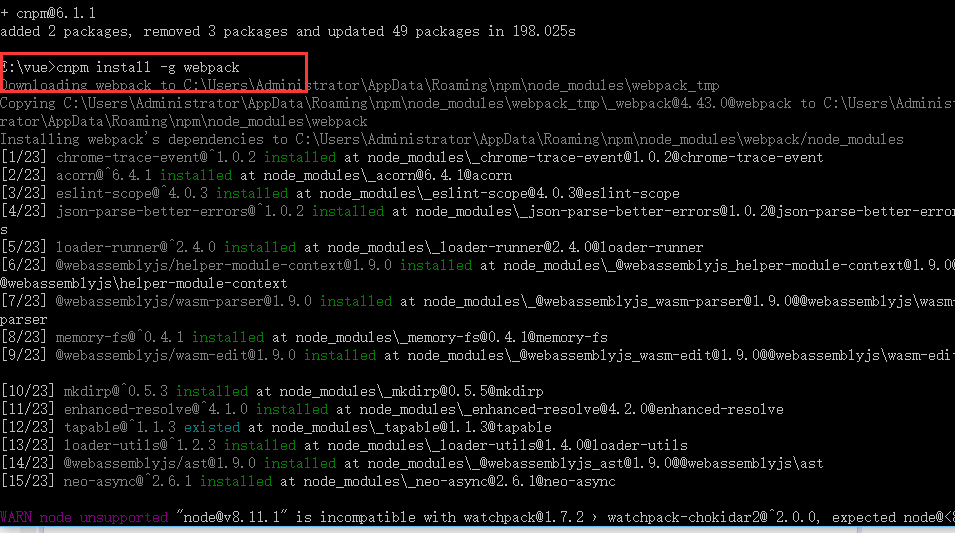
|  |
| --- |
| 1. **安装全局安装 vue-cli2**   如果需要用cli2 webpack构建项目，则必须安装webpack，否则不用  cnpm install -g webpack  cnpm install vue-cli -g  **2.安装全局安装 vue-cli3.0，用下列命令（我们用的）**  如果之前安装了2.0版本，先卸载 npm uninstall -g vue-cli )  npm uninstall vue-cli -g  cnpm install -g @vue/cli  **我们现在使用的是vue-cli3.0** |



这里的“-g”是全局安装，其目的是为了构建vue项目（分别安装）

安装后，输入 vue- V 查看vue版本号：若是出现，则说明安装成功。





参考https://www.cnblogs.com/coober/p/10875647.html

卸载：

cnpm uninstall -g @vue/cli

### 2.3 构建项目

构建项目有两种不同的方式，分别是cli2和cli3.0，我们使用的是cli3.0

#### 1. vue-cli2 创建项目

用Vue脚手架初始化一个基于 webpack 模板的新项目

假设我们要开发一个“后台管理”的项目，项目名称是“admin”

在CMD中运行命令：

|  |
| --- |
| vue init webpack admin |



这里，我没有开启ESLint，单元测试也暂不启用。

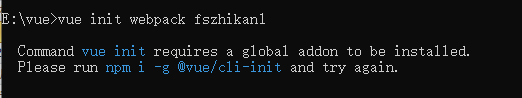
出现提示运行命令，就说明已经构建成功了。

npm install命令是：安装所需要的依赖；

npm run dev命令是：以开发模式启动项目；

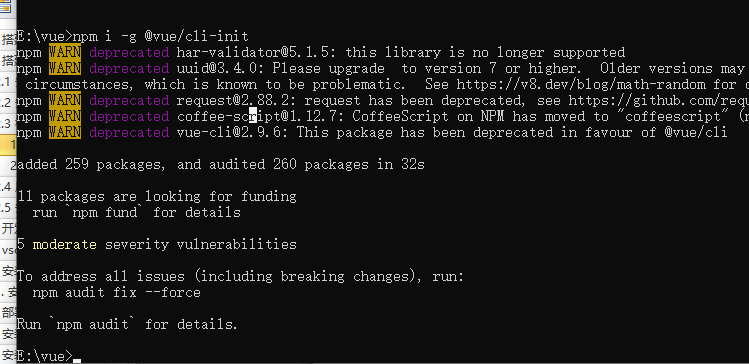
npm run build命令是：以生产模式生成静态页面文件。

如果在vue-cli2下错误，报以下错误



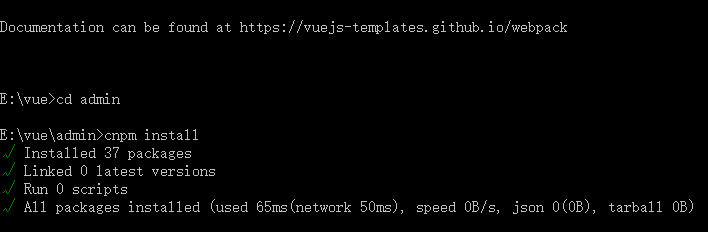
则安装npm i -g @vue/cli-init

npm i -g @vue/cli-init



1. 然后，进入文件夹，安装依赖：

|  |
| --- |
| cd admin  cnpm install |



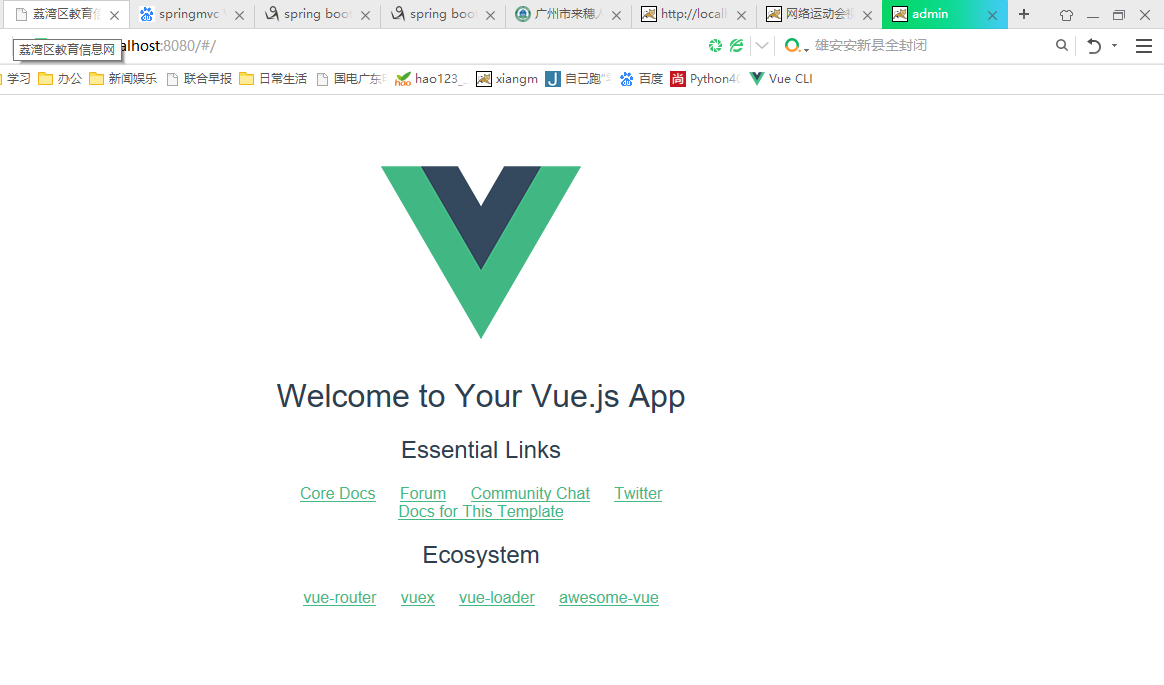
1. npm run dev 命令来启动项目

必须在项目目录下：

|  |
| --- |
| npm run dev |

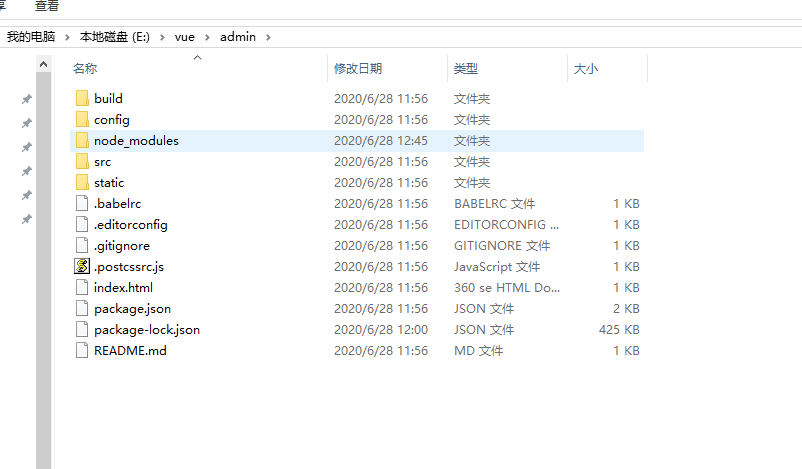


1. 这时，提示打开浏览器，访问<http://localhost:8080>



ok，出现大大的“V”字就说明项目已经构建完毕了。

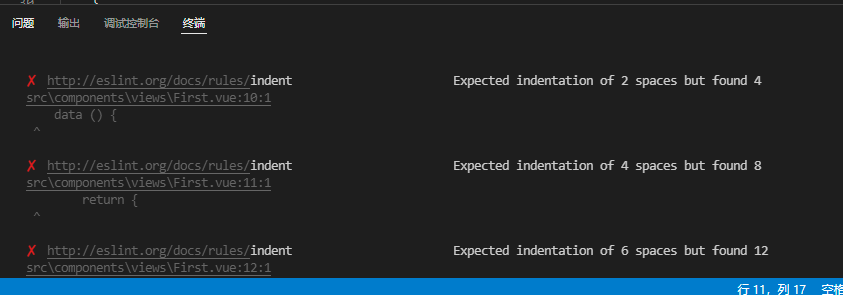
**在启动命令窗口中，按ctrl+c退出。**



控制台出现型如

✘ http://eslint.org/docs/rules/indent Expected indentation of 2 spaces but found 4

src\components\views\First.vue:10:1



错误缘由

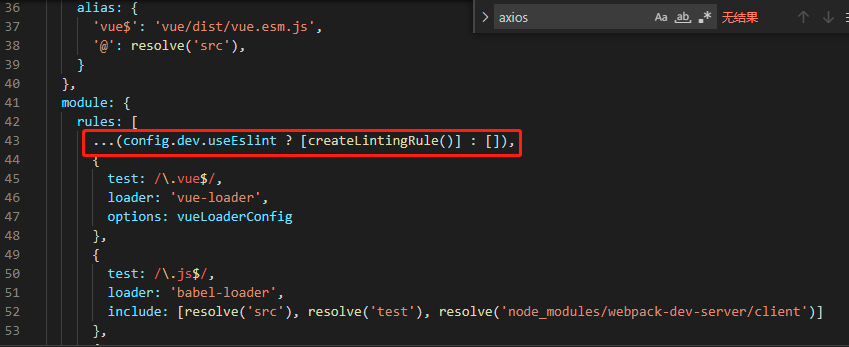
就是eslint 语法的检查功能，检查vue文件中 如果包含以下 注释信息，就会在终端输入这样的错误信息：

// 这是注释的内容信息

既然我们找到了问题的所在，那么如何在项目中关闭这个 eslint 的语法检查功能呢

解决办法

在 build/webpack.base.conf.js 文件中找到



将这个地方的代码注释删除都可以

之后重新 build 项目 ,启动就可以了

npm run build  
npm run dev

至此这个问题解决了

原文链接：<https://blog.csdn.net/qq_20667511/article/details/108308696>

#### 2. vue-cli3 创建项目（我们现在用的）

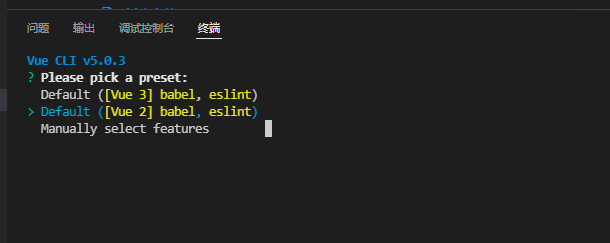
1）创建项目

vue create <Project Name> //文件名 不支持驼峰（含大写字母）

例如 vue create fszhikan

首先，会提示你选择一个preset（预设）：

① 除最后两个**，**其他选项都是你此前保存的预设配置（如下图第一个“ my-default”是我之前保存的预设配置，如今便可以直接用了）：



按按方向键 ↓选择，如果有保存过的，这里还会有已经保存过的预设，预设如下：

default（babel，eslint）：默认设置（直接enter）非常适合快速创建一个新项目的原型，没有带任何辅助功能的 npm包

 Manually select features：自定义配置（按方向键 ↓）是我们所需要的面向生产的项目，提供可选功能的 npm 包

**接下来（**Manually select features**）配置手动选择需要添加的配置项：**

手动选择需要添加的配置项：（上下键选择，空格键确定）

? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to invert selection)

>( ) Babel //转码器，可以将ES6代码转为ES5代码，从而在现有环境执行。

( ) TypeScript// TypeScript是一个JavaScript（后缀.js）的超集（后缀.ts）包含并扩展了 JavaScript 的语法，需要被编译输出为 JavaScript在浏览器运行，目前较少人再用

( ) Progressive Web App (PWA) Support// 渐进式Web应用程序

( ) Router // vue-router（vue路由）

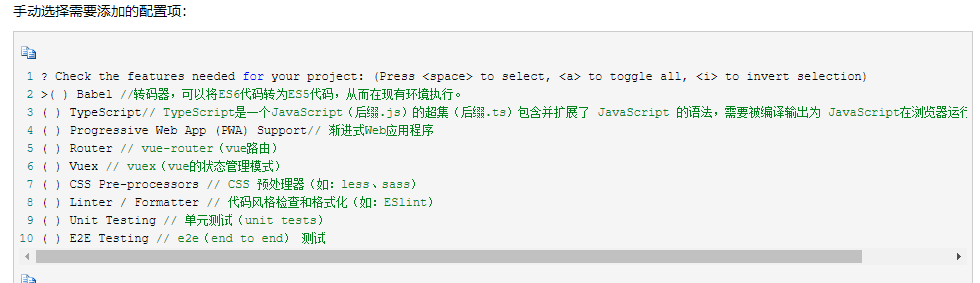
( ) Vuex // vuex（vue的状态管理模式）

( ) CSS Pre-processors // CSS 预处理器（如：less、sass）

( ) Linter / Formatter // 代码风格检查和格式化（如：ESlint）

( ) Unit Testing // 单元测试（unit tests）

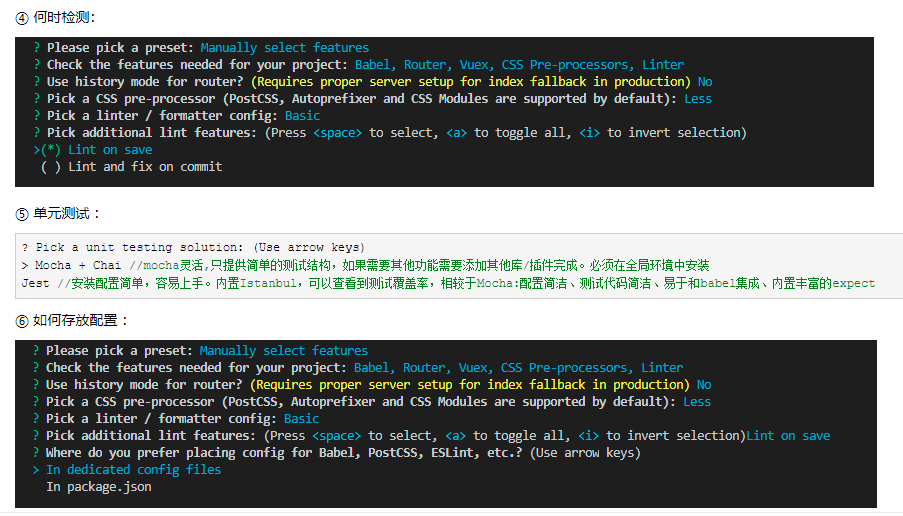
( ) E2E Testing // e2e（end to end） 测试

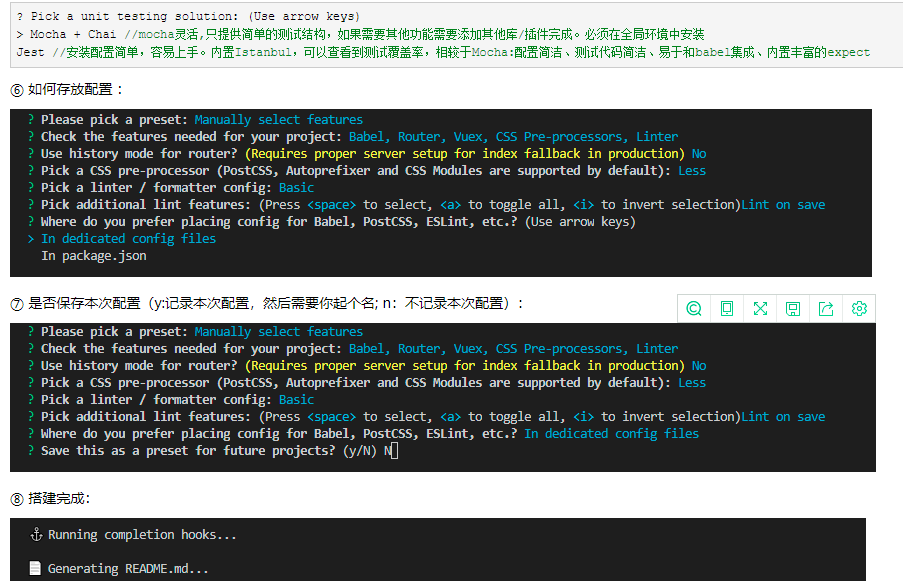


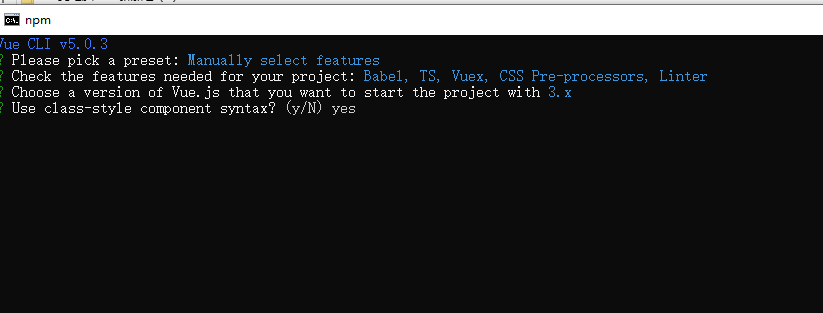
选择完后直接enter，然后会提示你选择对应功能的具体工具包，选择自己擅长或者使用广泛的（方便遇到问题时百度），简介如下：











### 2.4 启动项目

启动之前，如果没有依赖，需要安装

npm install

在项目目录下

vue-cli2启动：

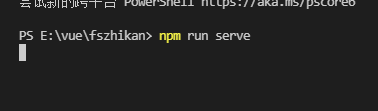
npm run dev

vue-cli3启动：

npm run serve

或者

yarn run serve



编译

yarn run lint

npm install antd babel-plugin-import *--save*

### 2.5 安装yarn

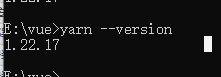
1．安装

npm install --global yarn



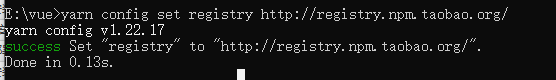
2验证

yarn --version



3设置淘宝镜像

yarn config set registry <http://registry.npm.taobao.org/>



## 三、安装插件

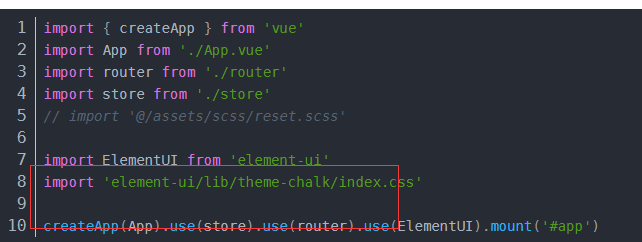
### 3.1 Element UI

使用的是最新的[Vue3.0](https://so.csdn.net/so/search?q=Vue3.0&spm=1001.2101.3001.7020" \t "_blank)，按照正常的elementui方式引入，结果页面出现空白，控制台出现报错情况。经了解，原来是从Vue3.0不支持Element UI，而是改成了Element Plus。

**Vue2.0安装**

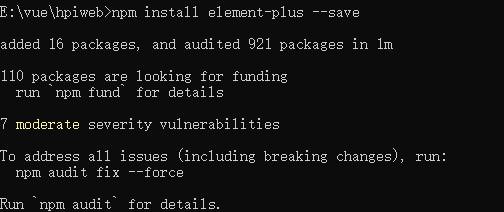
npm i element-ui -S

main.js中引入



**Vue3.0安装**

npm install element-plus --save





### 3.2 ant-design-vue

ant-design-vue是蚂蚁金服 Ant Design 官方唯一推荐的Vue版UI组件库，它其实是Ant Design的Vue实现，组件的风格与Ant Design保持同步，组件的html结构和css样式也保持一致

推荐使用 npm 或 yarn 的方式进行开发，不仅可在开发环境轻松调试，也可放心地在生产环境打包部署使用，享受整个生态圈和工具链带来的诸多好处。

$ npm install ant-design-vue –save

或者

$ yarn add ant-design-vue

推荐 yarn，npm容易报错

加入图标

npm install --save @ant-design/icons-vue

## 三、开发环境 VC CODE

VSCode是微软出的一款轻量级代码编辑器，免费而且功能强大，对JavaScript和NodeJS的支持非常好，自带很多功能，例如代码格式化，代码智能提示补全、Emmet插件等。VSCode推荐一个项目以文件夹的方式打开。

1.vscode安装

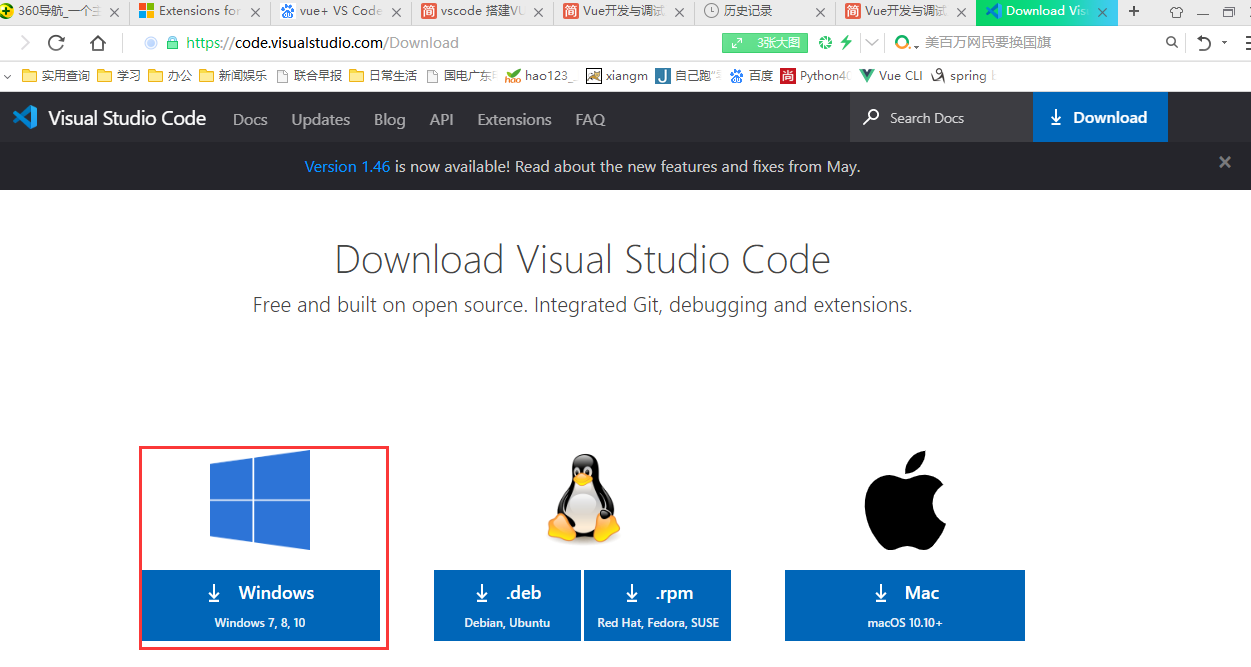
进入vscode官网（<https://code.visualstudio.com/Download）https://code.visualstudio.com/>

太慢的话，复制链接地址，如下：

<https://az764295.vo.msecnd.net/stable/899d46d82c4c95423fb7e10e68eba52050e30ba3/VSCodeUserSetup-x64-1.63.2.exe>

修改为如下国内地址

https://vscode.cdn.azure.cn/stable/899d46d82c4c95423fb7e10e68eba52050e30ba3/VSCodeUserSetup-x64-1.63.2.exe

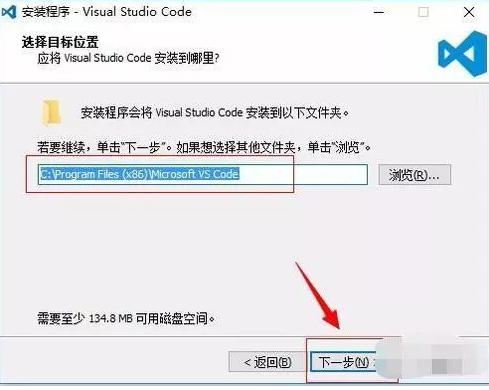


双击下载的安装包，进入安装向导界面；点击下一步

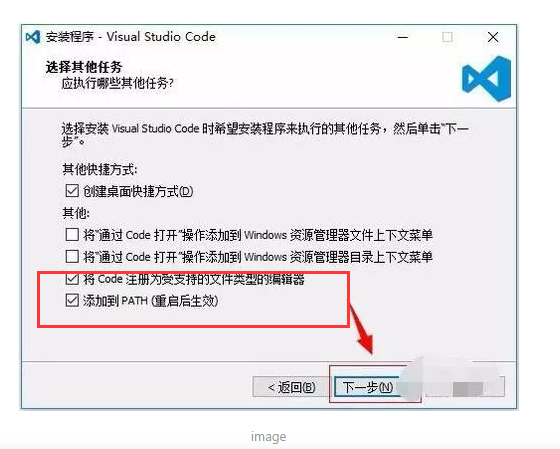


进入许可协议，点击“我接受协议”选项；点击下一步





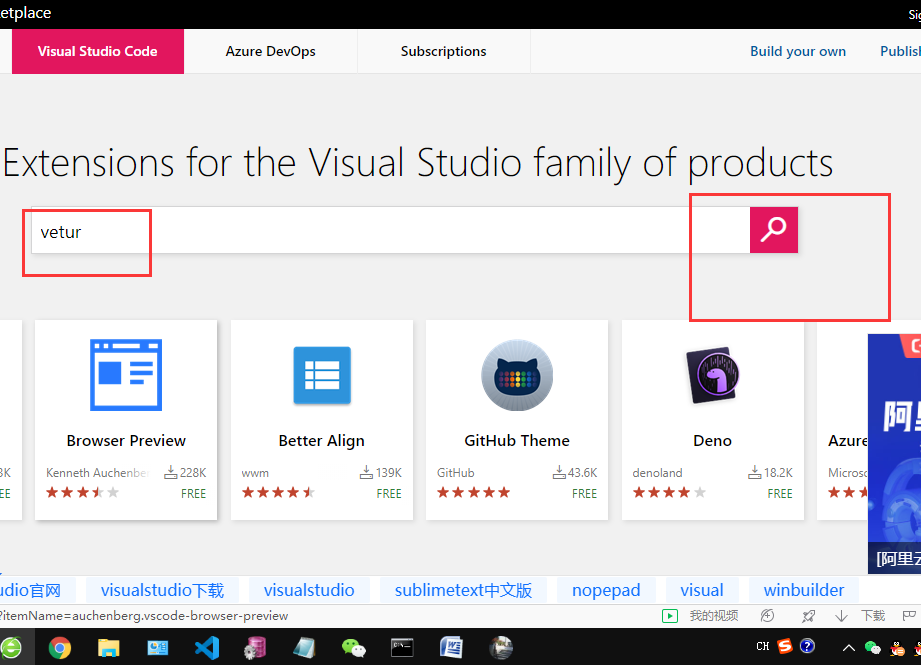
进入选择其他任务界面，默认已经勾选了必要的任务，这里尽量不要修改默认的，同时你也可以选择将打开方式添加到鼠标右键菜单上（勾选“其他”中的第一选项），继续点击下一步

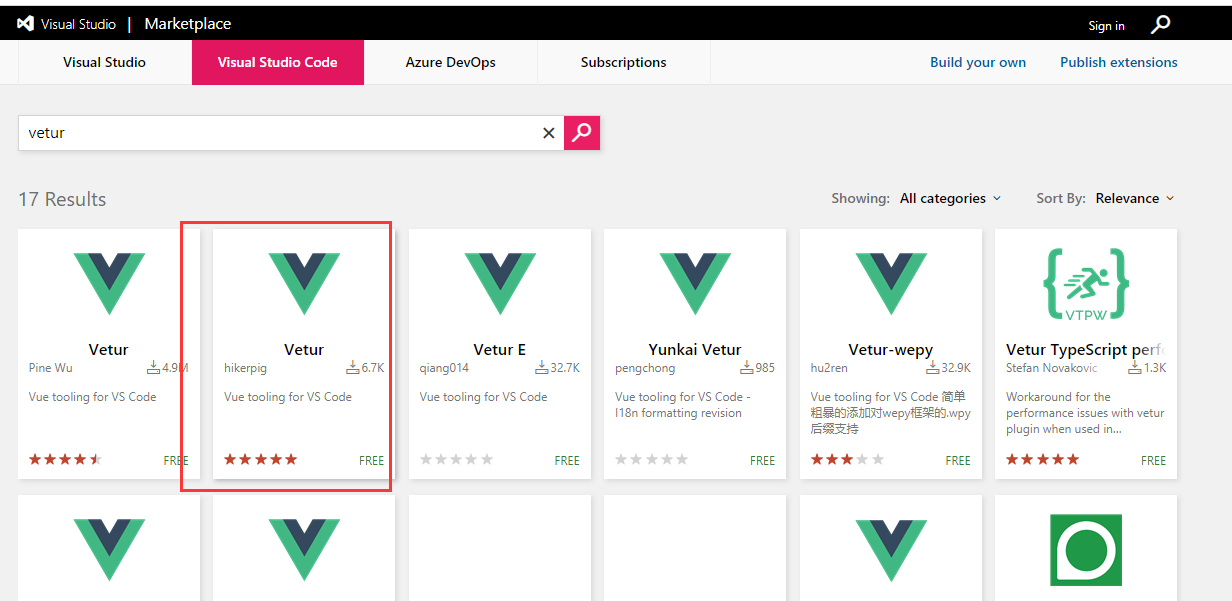


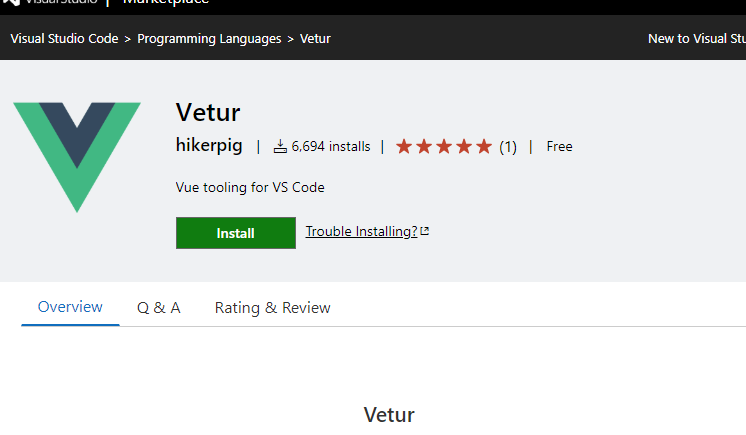
2.vscode插件

安装

进入vscode官网插件商店（https://marketplace.visualstudio.com/VSCode）







点击install,会启动vscode

安装就可以了

vue相关插件

vetur 语法高亮、智能感知

VueHelper vue代码碎片

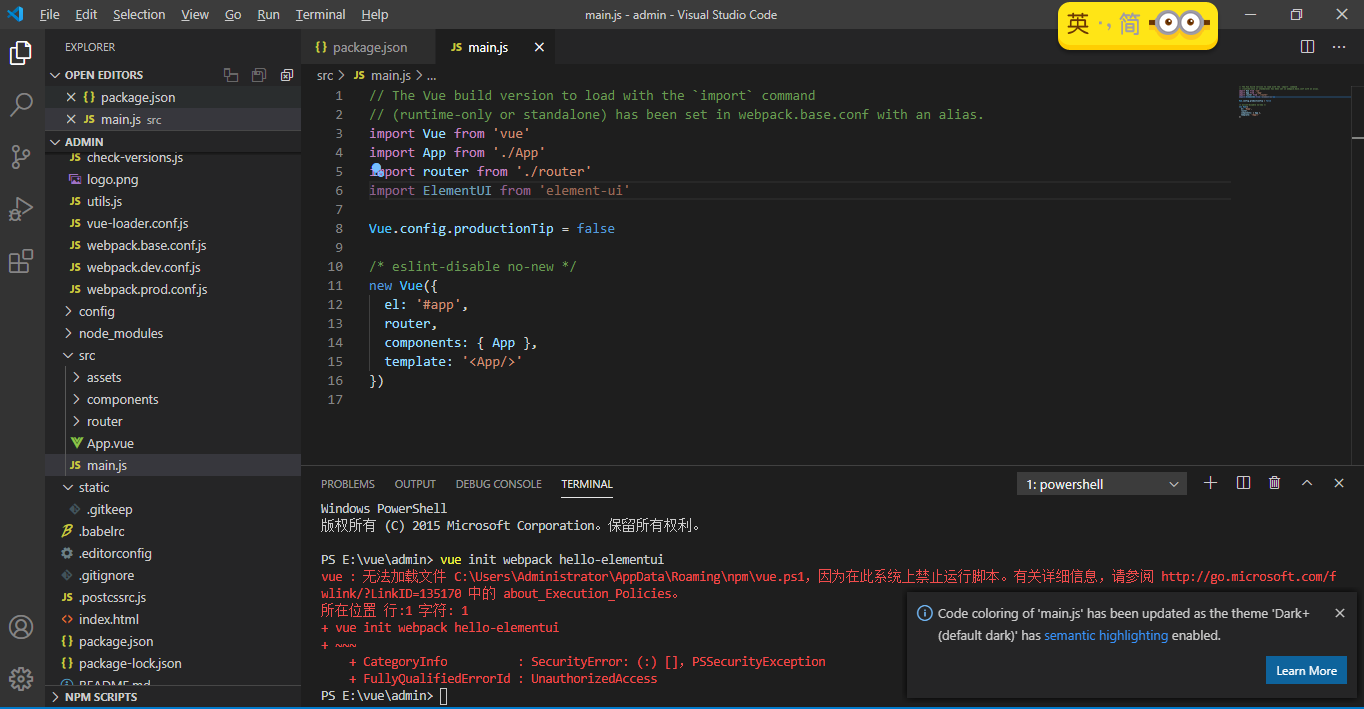
Vue 2 Snippets vue2代码碎片

导入工程

以vue项目为例

在vscode ide中选择"文件"-->"打开文件夹"

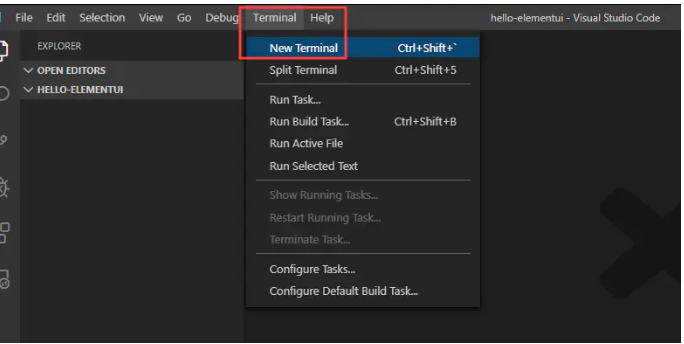
将构建好的vue工程导入



## 四、vscode 搭建VUE+element ui 项目

安装好node，npm以及cnpm .

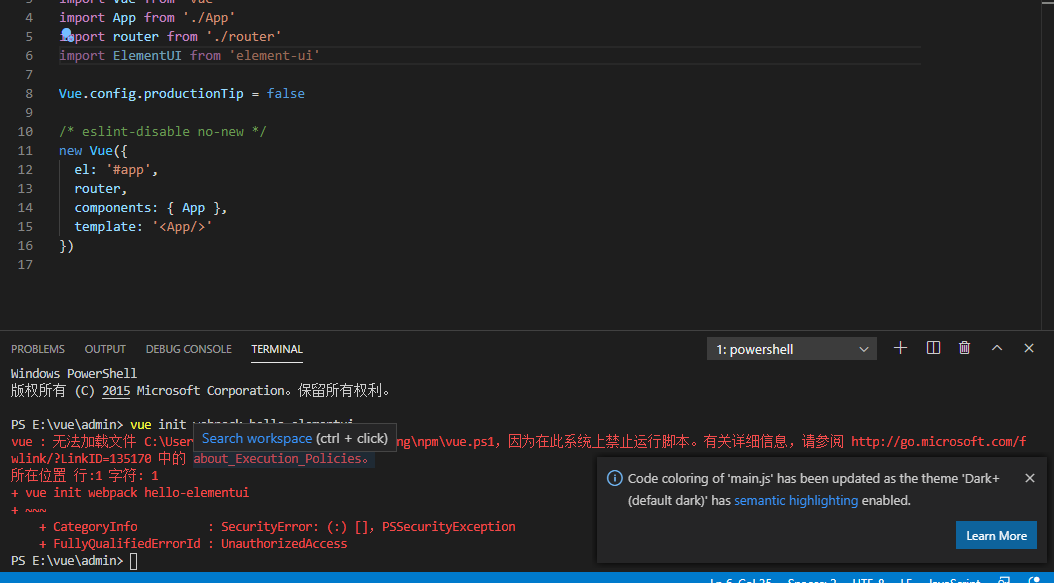
1. 点击vscode的terminal，打开一个新的命令行窗口。



新建项目（可以在cmd下创建）

vue init webpack hello-elementui

如果出现如下错误：  
vue : 无法加载文件 D:\Program Files\nodejs\node\_global\vue.ps1，因为在此系统上禁止运行  
脚本。有关详细信息，请参阅 https:/go.microsoft.com/fwlink/?LinkID=135170 中的 about\_Exe  
cution\_Policies。  
所在位置 行:1 字符: 1  
vue init webpack hello-elementui  
CategoryInfo : SecurityError: (:) []，PSSecurityException  
FullyQualifiedErrorId : UnauthorizedAccess



进行如下操作





1. 添加element-ui

方法1：（建议这种，安装最新的）

第一步：安装element-ui，运行

npm i element-ui –S

，注意必须在项目下（和package.json文件目录平行）

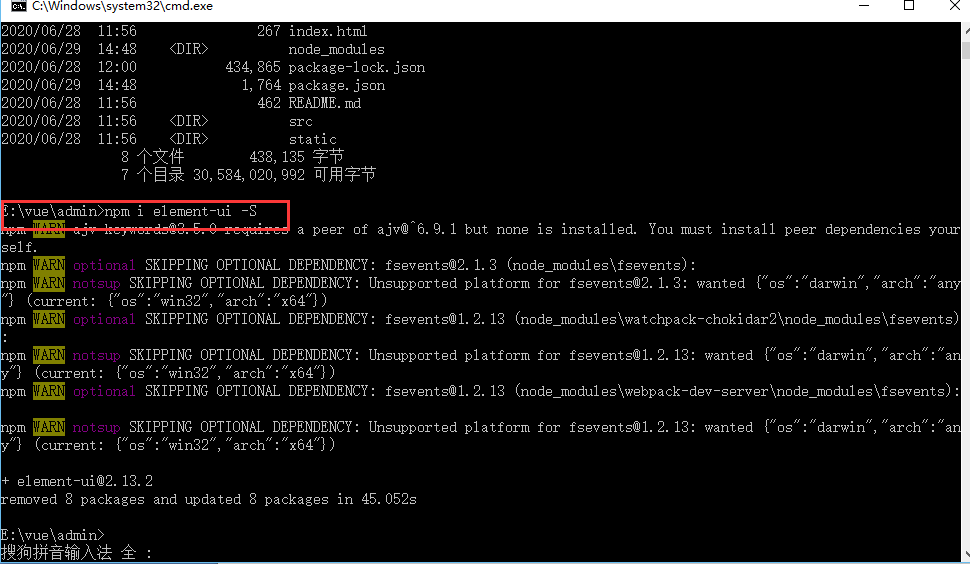
安装前注意把npm/yarn设置镜像服务器（因为大部分内容要去外网下载，直连比较慢）

NPM设置成淘宝镜像

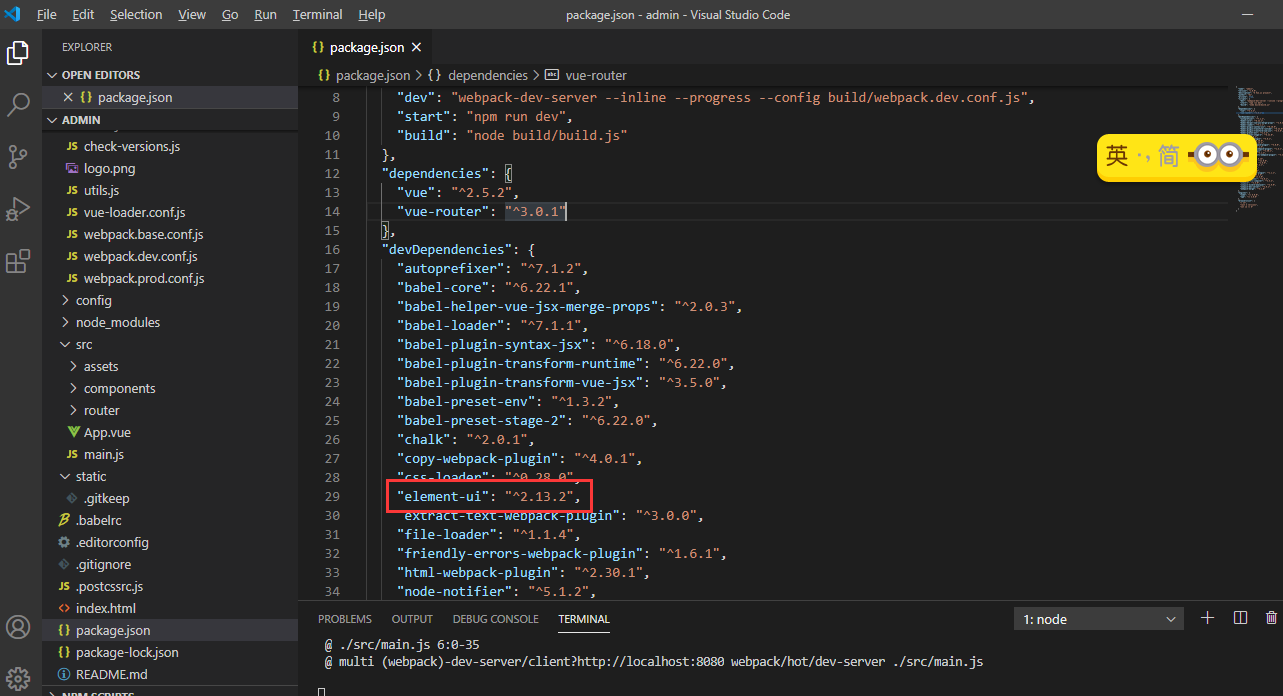
npm config set registry http://registry.npm.taobao.org/ -g

yarn设置成淘宝镜像

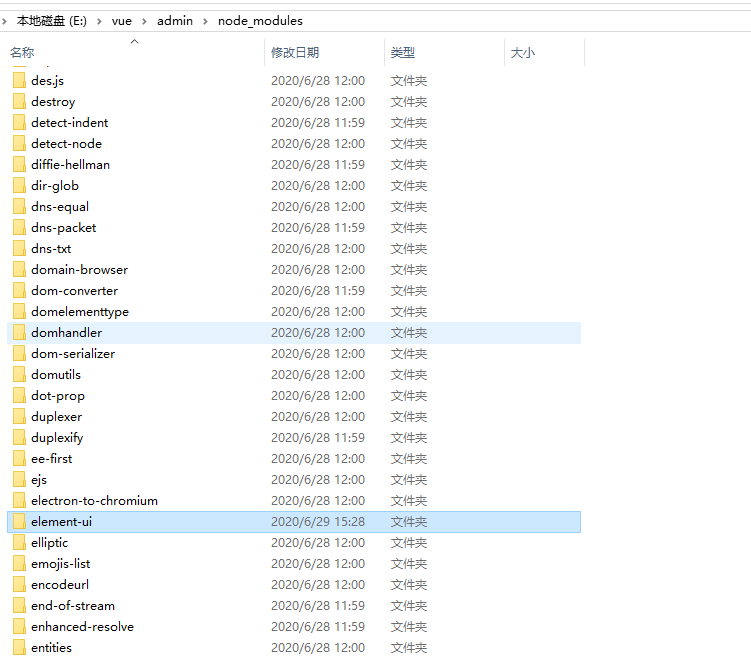
yarn config set registry http://registry.npm.taobao.org/ -g



安装完成后查看package.json，显示如下：



在项目目录下如下：



第二部：在项目里面引用了elemtnt-ui，打开src目录下的main.js



方法2：

打开项目根目录下的package.json文件，在devDependencies节点下添加element-ui的依赖：

{

"name": "hello-elementui",

"version": "1.0.0",

"description": "a vue project",

"author": "mcmin001 <mcmin001@sina.com>",

"private": true,

"scripts": {

"dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",

"start": "npm run dev",

"lint": "eslint --ext .js,.vue src",

"build": "node build/build.js"

},

"dependencies": {

"vue": "^2.5.2",

"vue-router": "^3.0.1"

},

"devDependencies": {

"autoprefixer": "^7.1.2",

"babel-core": "^6.22.1",

"babel-eslint": "^8.2.1",

"babel-helper-vue-jsx-merge-props": "^2.0.3",

"babel-loader": "^7.1.1",

"babel-plugin-syntax-jsx": "^6.18.0",

"babel-plugin-transform-runtime": "^6.22.0",

"babel-plugin-transform-vue-jsx": "^3.5.0",

"babel-preset-env": "^1.3.2",

"babel-preset-stage-2": "^6.22.0",

"chalk": "^2.0.1",

"copy-webpack-plugin": "^4.0.1",

"css-loader": "^0.28.0",

"eslint": "^4.15.0",

"eslint-config-standard": "^10.2.1",

"eslint-friendly-formatter": "^3.0.0",

"eslint-loader": "^1.7.1",

"eslint-plugin-import": "^2.7.0",

"eslint-plugin-node": "^5.2.0",

"eslint-plugin-promise": "^3.4.0",

"eslint-plugin-standard": "^3.0.1",

"eslint-plugin-vue": "^4.0.0",

"extract-text-webpack-plugin": "^3.0.0",

"file-loader": "^1.1.4",

"friendly-errors-webpack-plugin": "^1.6.1",

"html-webpack-plugin": "^2.30.1",

"node-notifier": "^5.1.2",

"optimize-css-assets-webpack-plugin": "^3.2.0",

"ora": "^1.2.0",

"portfinder": "^1.0.13",

"postcss-import": "^11.0.0",

"postcss-loader": "^2.0.8",

"postcss-url": "^7.2.1",

"rimraf": "^2.6.0",

"semver": "^5.3.0",

"shelljs": "^0.7.6",

"uglifyjs-webpack-plugin": "^1.1.1",

"url-loader": "^0.5.8",

"vue-loader": "^13.3.0",

"vue-style-loader": "^3.0.1",

"vue-template-compiler": "^2.5.2",

"webpack": "^3.6.0",

"webpack-bundle-analyzer": "^2.9.0",

"webpack-dev-server": "^2.9.1",

"webpack-merge": "^4.1.0",

"element-ui":"^2.2.1"

},

"engines": {

"node": ">= 6.0.0",

"npm": ">= 3.0.0"

},

"browserslist": [

"> 1%",

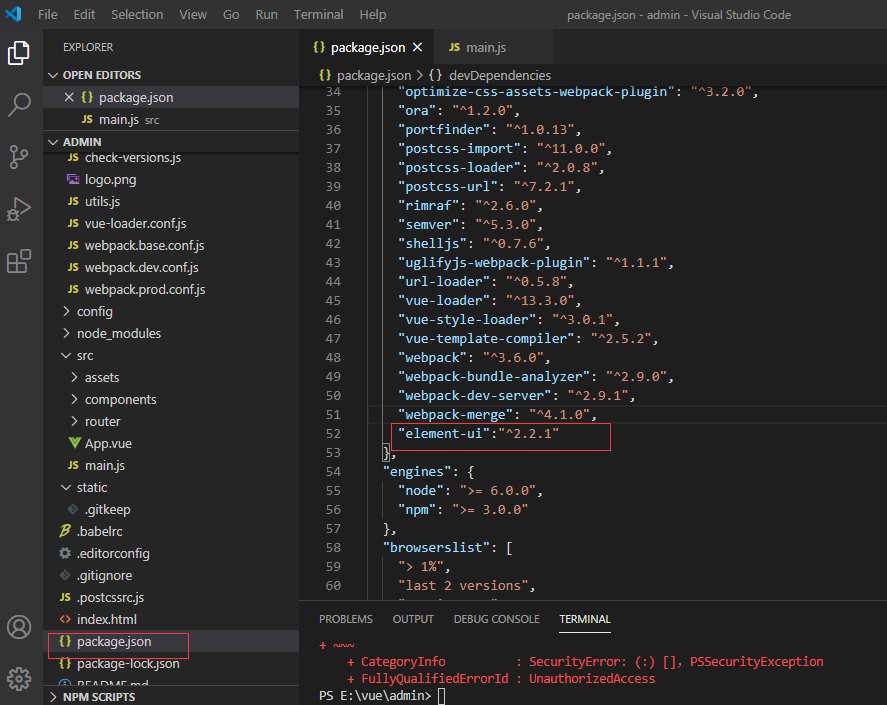
"last 2 versions",

"not ie <= 8"

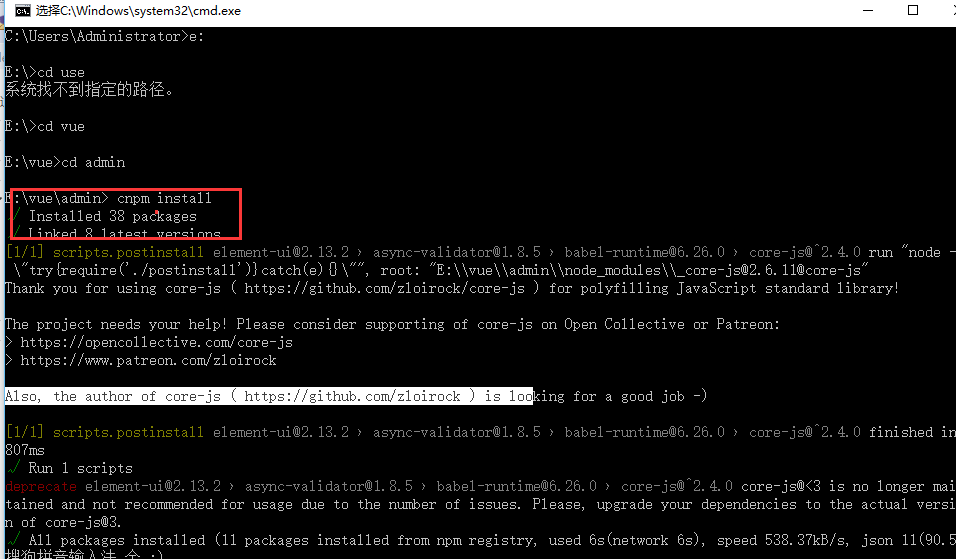
]

}

如下图

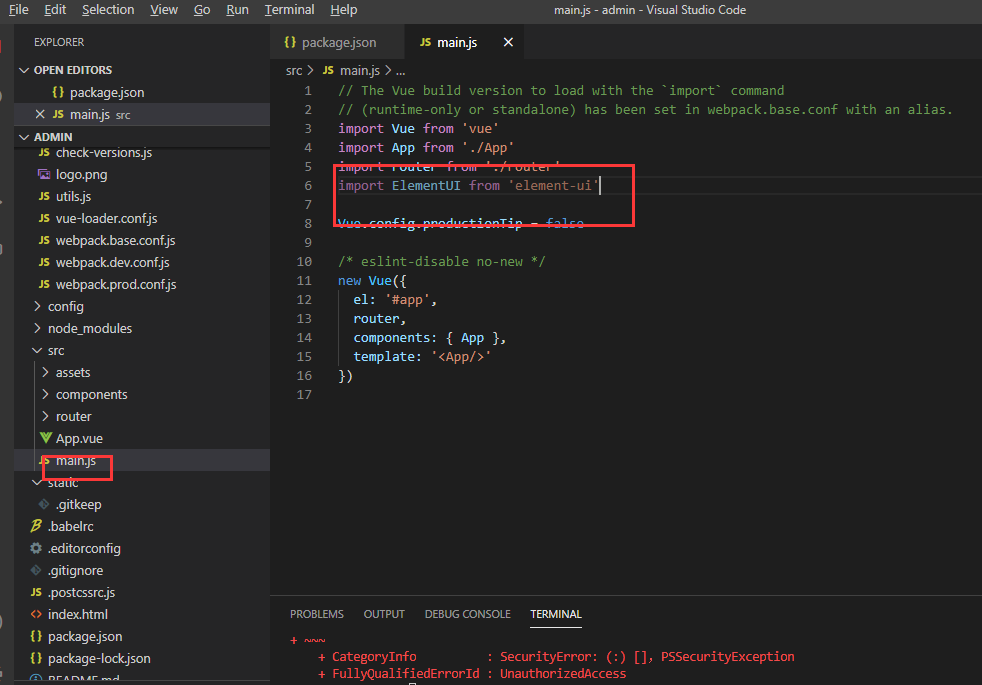


然后在命令行下执行 cnpm install



在main.js文件中添加element-ui的引用：

import ElementUI from 'element-ui'



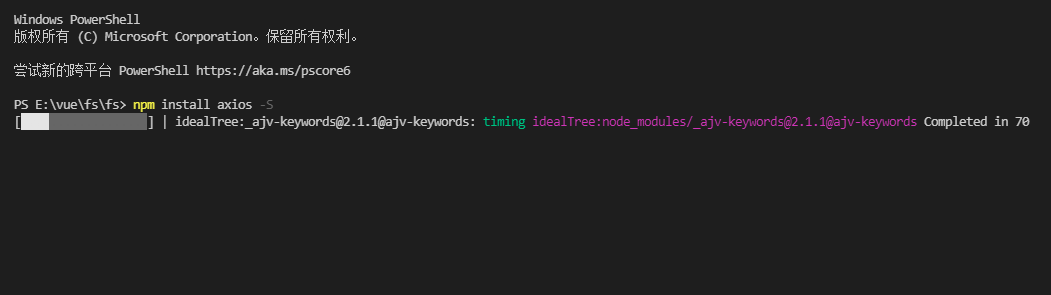
## 四、安装插件

### 安装axios

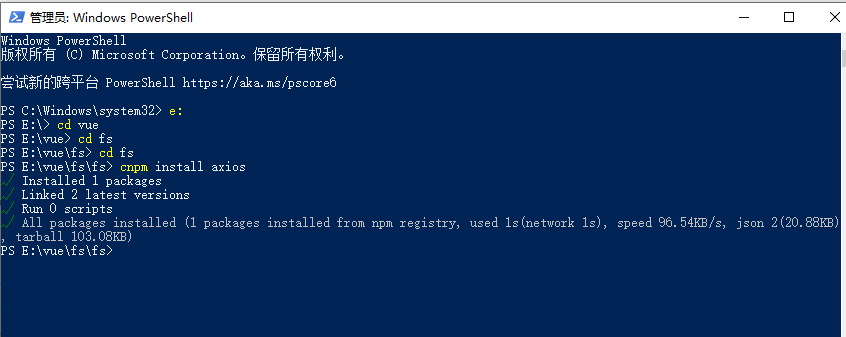
axios是一个基于 promise 的 HTTP 库，在vue中axios是比较常用的网络请求方法。

安装

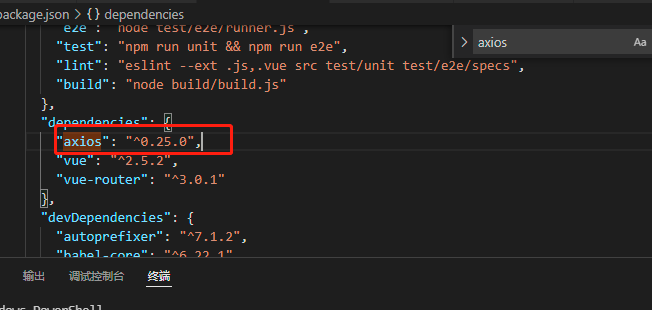
npm install axios -S



或者



安装完成以后，package.json 看看有没有axios的版本信息

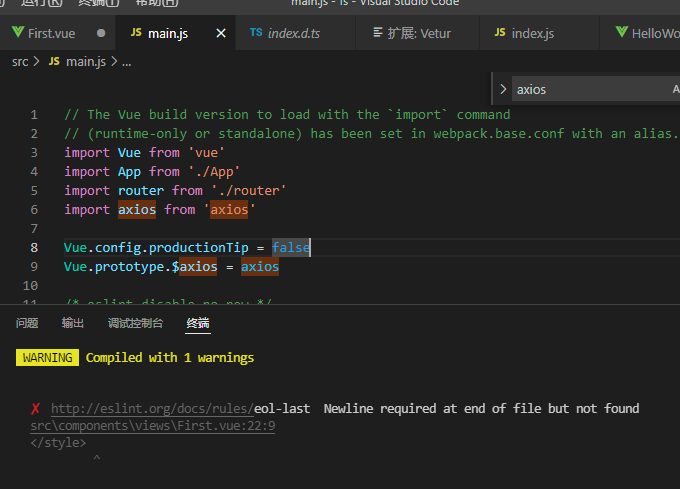


在项目中应用

在在main.js配置

import axios from 'axios'

Vue.prototype.$axios = axios



## 五、部署下载的项目

1. 下载

从Git上拉去代码了或者从网站上下载压缩包，下载下来如下图





1. 进入项目目录

E:\vue>cd vue-element-admin

1. 安装依赖

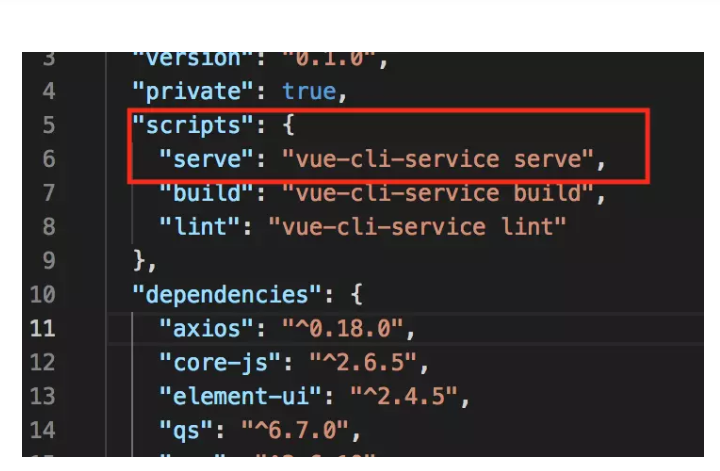
npm install

或者npm install --registry=https://registry.npm.taobao.org (解决网速比较慢的问题)

1. 启动

npm run dev

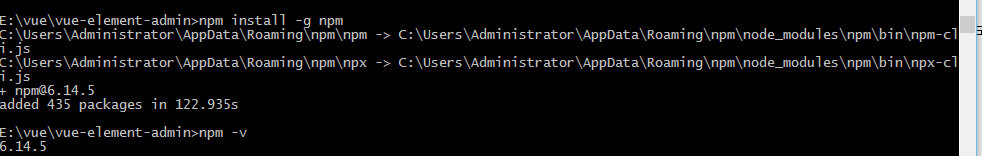
或者npm run serve



在script里，并没有dev，而是serve，应该用npm run serve命令运行vue项目即可。

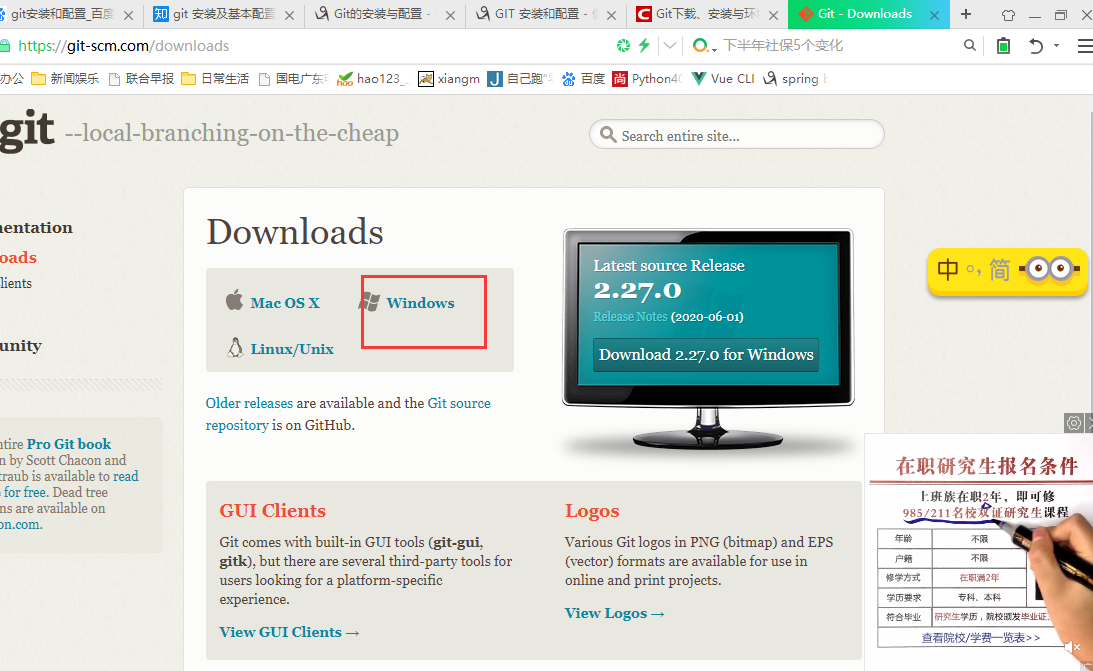
1. 升级npm

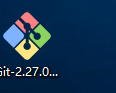
npm install -g npm



## 六、安装GIT

1. 下载 官方网站：<https://git-scm.com/downloads>

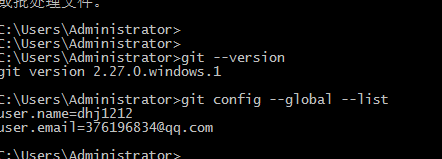




下载完成后基本上就是下一步下一步

1. 检查是否安装成功

git –version





1. 配置Git用户和邮箱

因为Git是分布式版本管理工具，所以每一个使用者需要提供个人信息，留下个人联系方式便于别的开发者联系。

配置用户

# 配置用户名

git config --global user.name "username" //（ "username"是自己的账户名，）

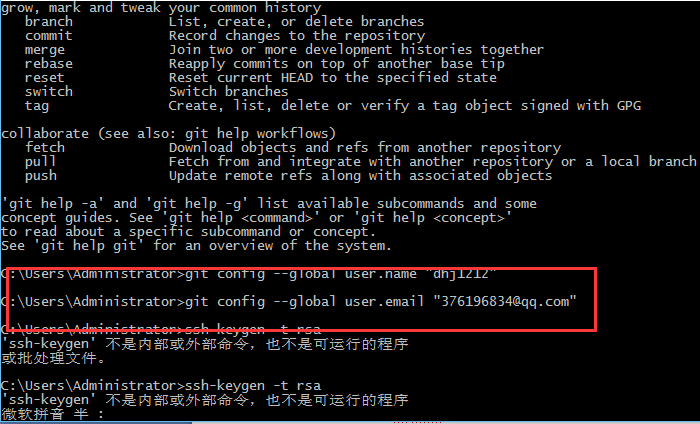
# 配置邮箱

git config --global user.email "username@email.com" //("username@email.com"注册账号时用的邮箱)

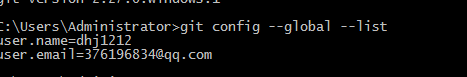
例如：

git config --global user.name "dhj1212"

git config --global user.email [376196834@qq.com](mailto:376196834@qq.com)



以上命令执行结束后，可用 git config --global --list 命令查看配置是否OK



1. 生成SSH密钥

SSH是Secure Shell安全外壳协议的缩写。它为网络服务提供目前较为可靠的安全协议，有效防止信息泄露问题。通过公钥和私钥的验证，建立安全连接。使用SSH方式和远程版本库通信的时候可以无需输入用户名密码进行验证。

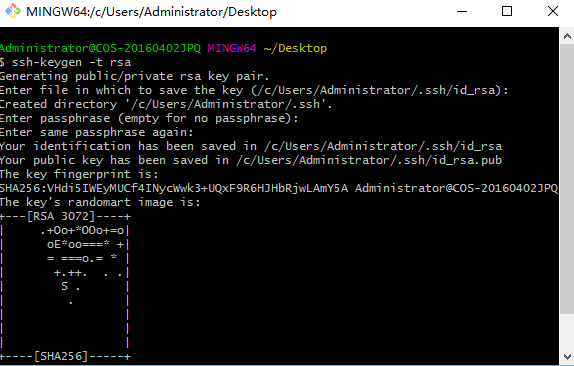
在任意空白位置右击选择"Git Bash Here"，启动Git的终端模拟器。

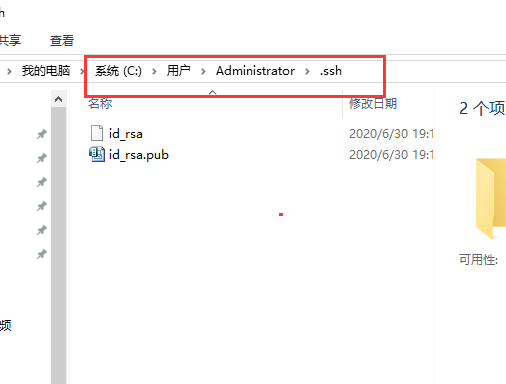
输入： ssh-keygen -t rsa 后回车，会依次要求输入。默认为空，三个空格即可

　　1) 密钥保存位置，默认是 /c/Users/用户名/.ssh/ 目录下。

　　2) 输入密码，默认无密码。

　　3) 确认输入密码，默认无密码。





## 六、安装Yarn

1.安装

Cmd下执行npm i yarn –g

npm i yarn –g

其中

-i：install

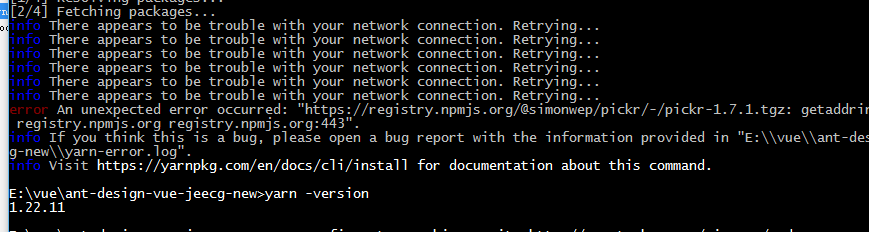
-g：全局安装（global）,使用 -g 或 --global

1. 验证

yarn -version



1. 解决**资源地址请求超时**问题



在使用yarn安装依赖包时报错，提示信息：error An unexpected error occurred: "https://registry.yarnpkg.com/killable/-/killable-1.0.0.tgz: connect ETIMEDOUT 104.16.19.35:443"

这是由于资源地址请求超时造成的，更换一下请求地址即可。

————————————————

版权声明：本文为CSDN博主「前端灯塔」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/pznavbypte/article/details/90710878>

1. 运行命令，把资源地址设置成npm淘宝源

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

npm config set disturl https://npm.taobao.org/dist

1. **附带问题：node-sass缓慢**

npm config rm proxy

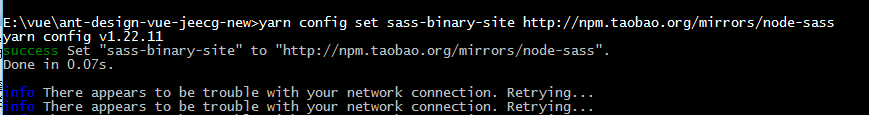
npm config rm https-proxy

1. 对node-sass镜像源进行设置

yarn config set sass-binary-site <http://npm.taobao.org/mirrors/node-sass>

或者 npm config set sass-binary-site <http://npm.taobao.org/mirrors/node-sass>

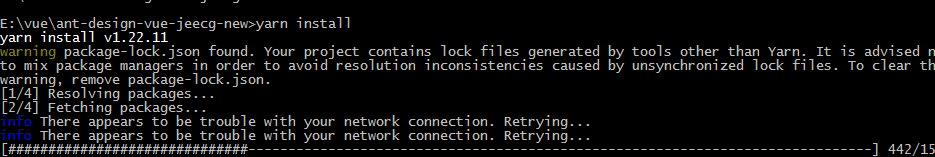
我用的npm



1. 安装依赖

必须在 工程目录下执行

yarn install



1. 常用命令

yarn / yarn install 等同于npm install 批量安装依赖

yarn add xxx 等同于 npm install xxx —save 安装指定包到指定位置

yarn remove xxx 等同于 npm uninstall xxx —save 卸载指定包

yarn add xxx —dev 等同于 npm install xxx —save-dev

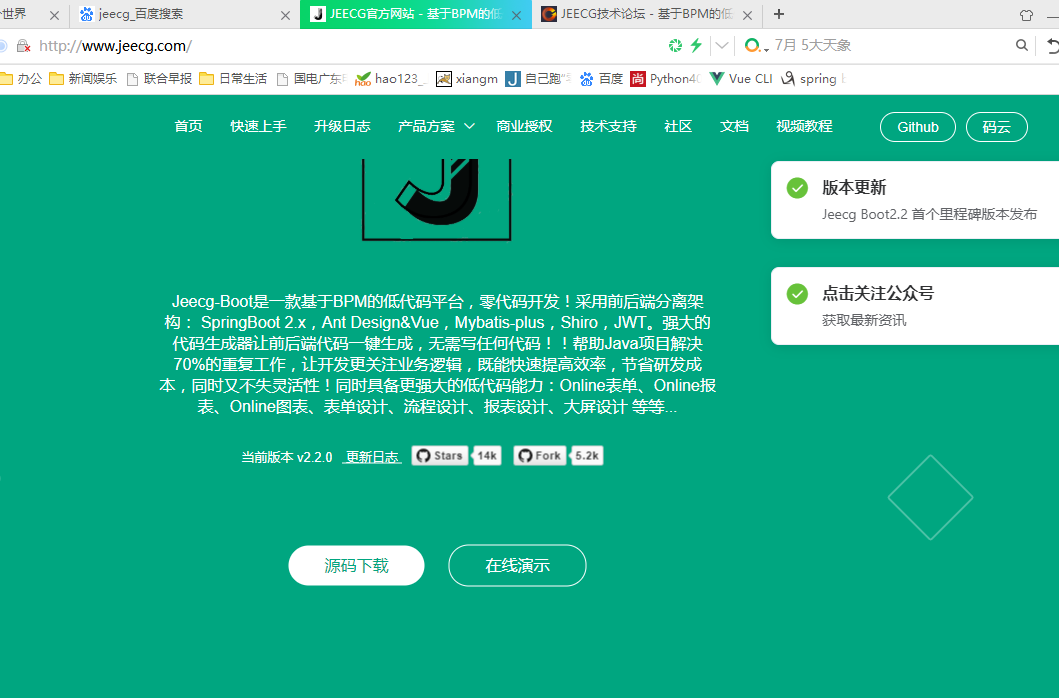
yarn upgrade 等同于 npm update 升级全部包

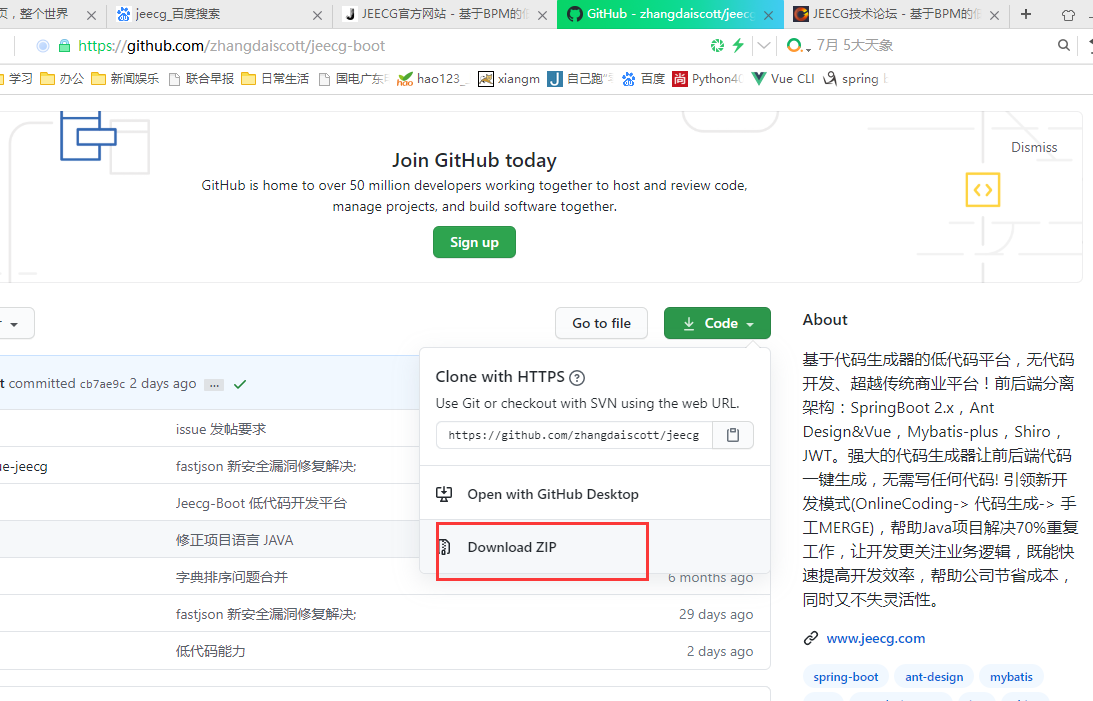
yarn global add xxx 等同于 npm install xxx -g 全局安装指定包

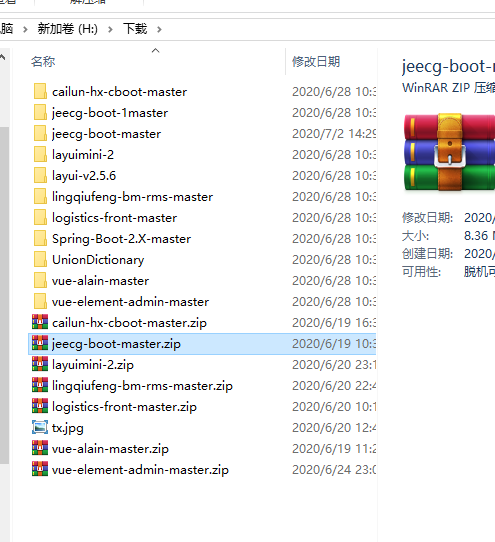
## 七、jeecg-boot部署

1、下载源码：

登陆<http://bbs.jeecg.com/>，下载最新的源码

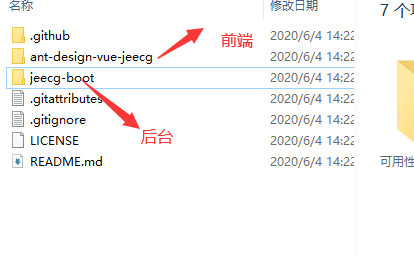


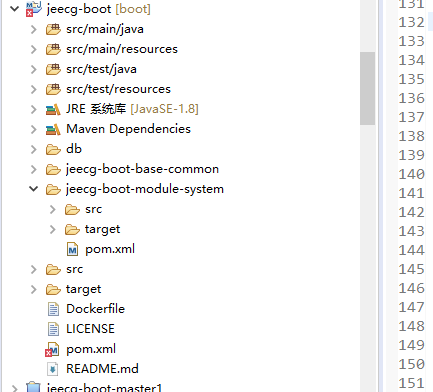




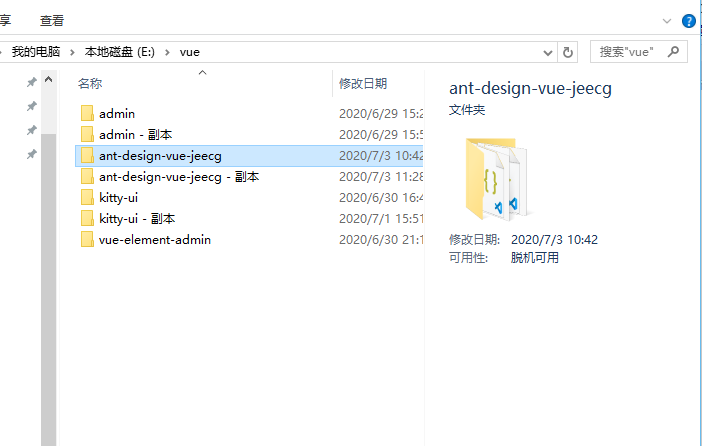
2、部署：

新建一个maven项目jeecg-boot，把下载的jeecg-boot-master解压，把jeecg-boot文件夹对应覆盖自己创建的jeecg-boot





把ant-design-vue-jeecg放在vue工程下安装依赖



然后安装依赖

npm install

例如：

E:\vue\ant-design-vue-jeecg> npm install

或者

安装完成后启动vue

E:\vue\ant-design-vue-jeecg> npm run serve

部署后台，

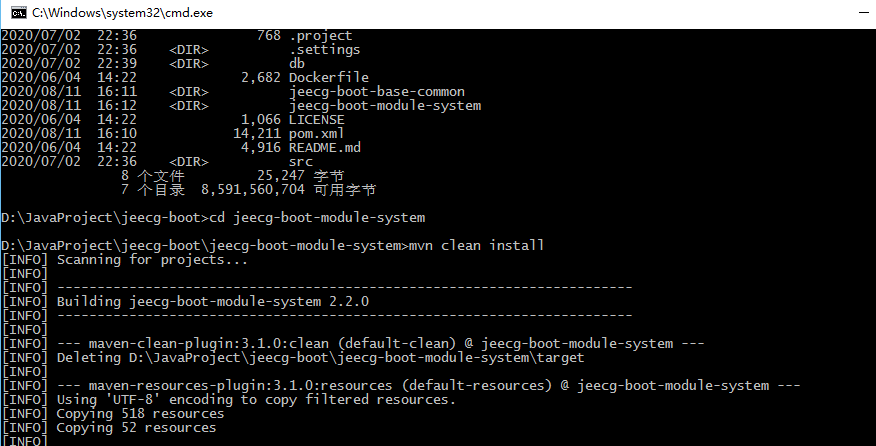
在创建的jeecg-boot，maven编译工程

在D:\JavaProject\jeecg-boot\jeecg-boot-module-system>mvn clean install 下编译

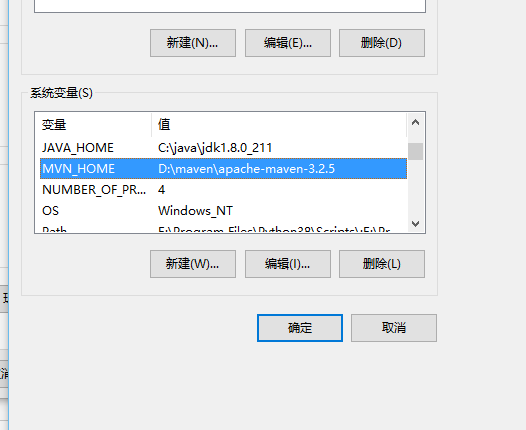
mvn clean install

或者

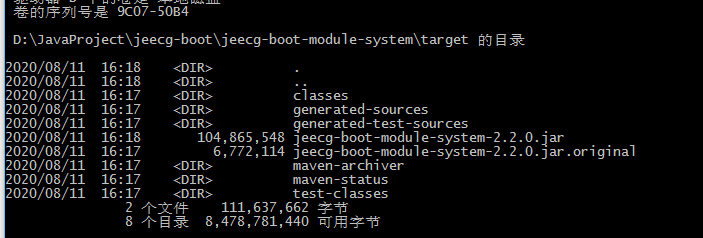
mvn clean package



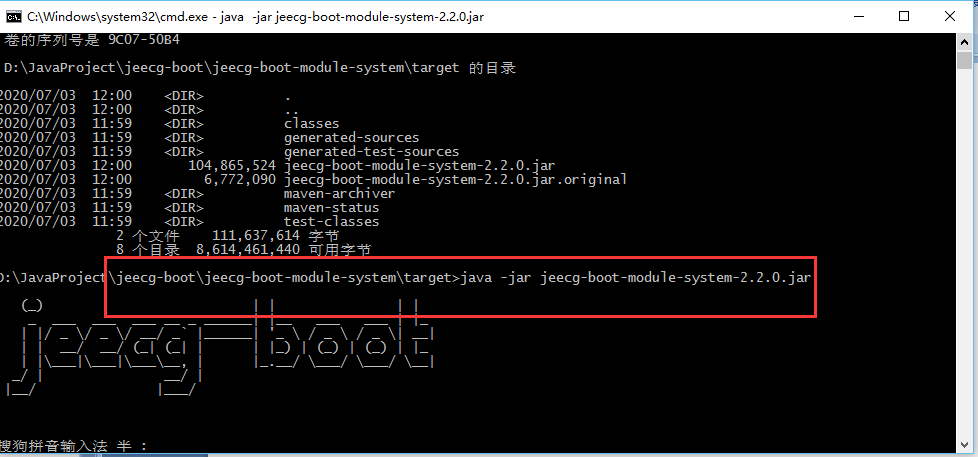
如果maven3.6.3编译有问题，请在环境变量改成3.2.5



编译成功后



启动如下：



3、创建数据库：

Mysql数据库

注意设置数据库不区分大小写

3、启动：

<http://localhost:3000>

登陆admin/123456

5、打包(在cmd下，进入工程目录)

需要注意的是

使用yarn安装node\_modules（注意！注意！注意！不能用其他方式，例如npm install） yarn install

Nodes 版本不能小于node-v12.13.1

步骤如下：

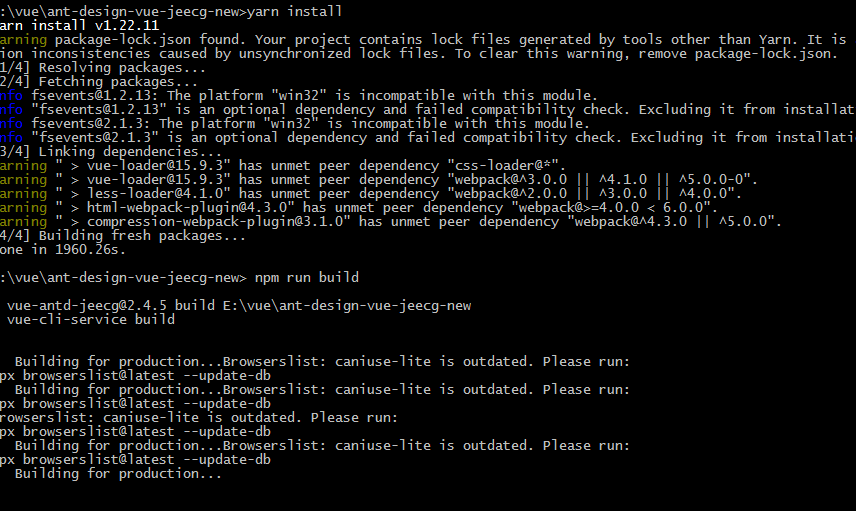
1. 先 yarn

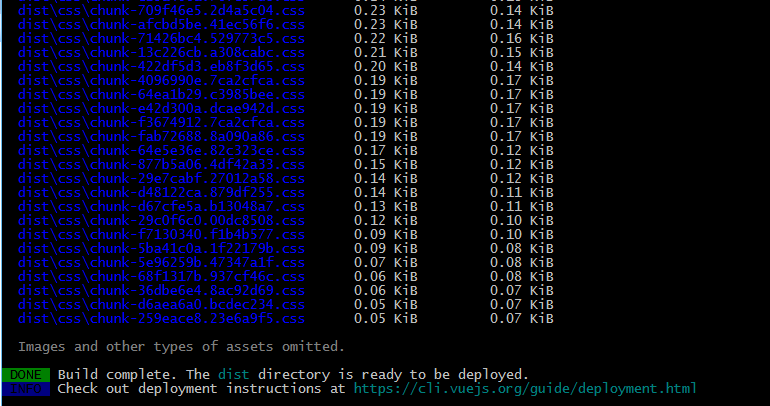
yarn install

1. build

npm run build

build 如果报错，将node\_modules文件删除,重新来一次



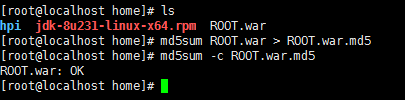


## 八、部署包生成MD5

1.在liunux下，上传部署包，例如：ROOT.WAR

2. 生成MD5文件，文件名称要一样

md5sum ROOT.war > ROOT.war.md5



3、后通过命令验证这个war包的MD5值是不是和它的MD5文件一致，目的是实施人员在实施前应用以下命令去校验MD5值，来衡量包是否完整

md5sum -c ROOT.war.md5 件和文件的MD5文件在一个目录下

