**论坛系统**

**开发环境搭建说明书**

**（2019.12.04）**

# 软件工具

## 基本工具

1. 开发环境

nodejs （建议版本为10以上）

1. 版本管理工具

Git-2.6.3-64-bit

TortoiseGit-2.6.0.0-64bit.msi

TortoiseGit-LanguagePack-2.6.0.0-64bit-zh\_CN.msi

1. 开发工具

VSCodeUserSetup-x64-1.40.2.exe

以上工具根据系统位数及版本需要，网上自行下载即可

# 工具安装

## 开发环境搭建

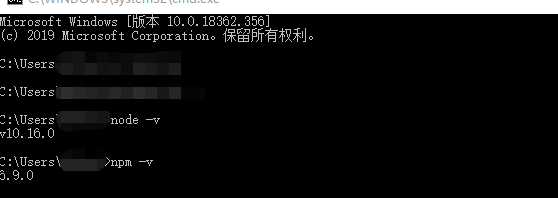
进入Node.js官网（<http://nodejs.cn/download/>）下载指定平台安装包（nodejs.org）;根据自己需要下载对应的版本,我下载的是windows系统64位的.msi版本。



点击安装文件，根据自己喜好选择安装路径，点击下一步直至完成

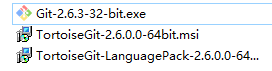


在cmd命令行中使用 node -v 或者node --version查看是否安装成功；显示版本信息证 明成功否则失败。新版本的nodejs安装的时候自动安装npm 使用npm -v查看；



## 版本工具Git安装及TortoiseGit使用

（1）首先安装相应的GIT工具，

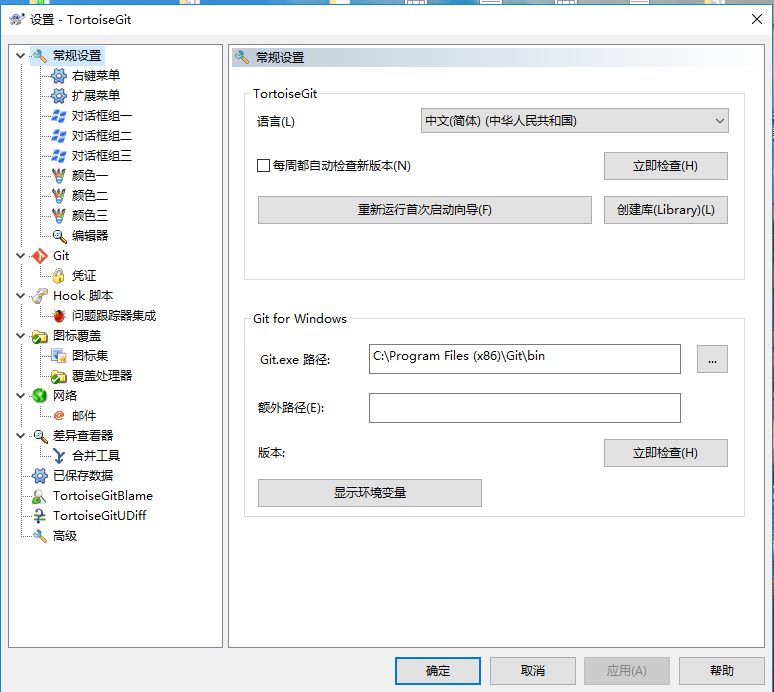


前两个为安装包，第三个为TortoiseGit的汉化包，安装过程基本都是直接下一步即可。

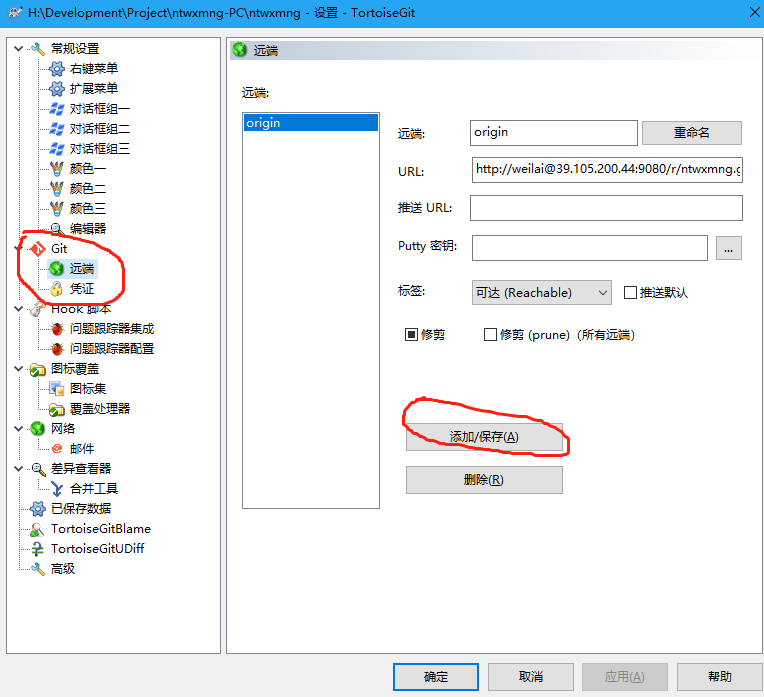
（2）远程仓库GitBlit,直接百度下载安装。

（3）通过本地TortoiseGit连接远程仓库 ，选择本地目录右键，TortoiseGit，选择设置

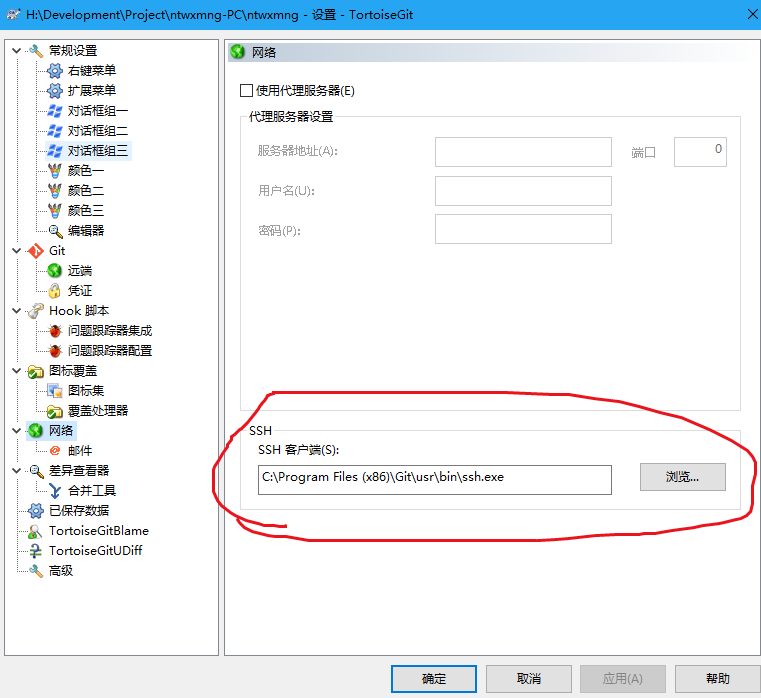




（4）点击远端，添加关联的远程仓库，地址在GitBlit版本库中复制



（5）选择 网络，设置好你本地Git目录下的执行文件



### 获取版本

（1）获取最新版本分两种方式，拉取（pull）和 获取（fetch）。

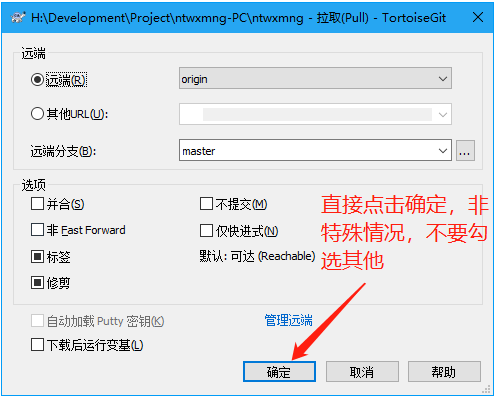
拉取，一般用于最新版本中有修改文件并且已存在的文件，获取时会与本地文件自动合并。

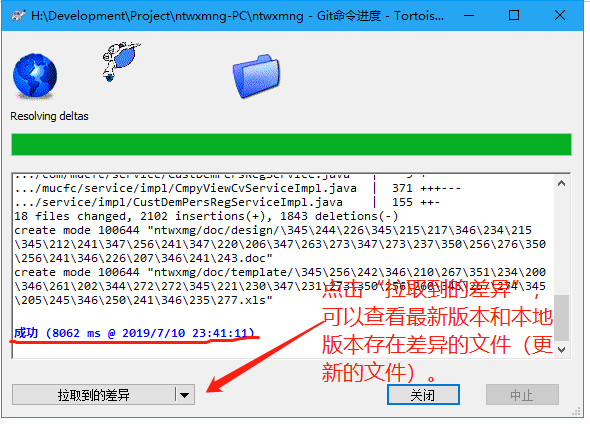
获取，一般用于获取的最新版中存在差异的文件是本地没有的，获取时不会自动合并，所以相对来说获取的方式较为安全，但出于本项目中的情况考虑，

**建议，直接使用 拉取（PULL）获取最新版本！**



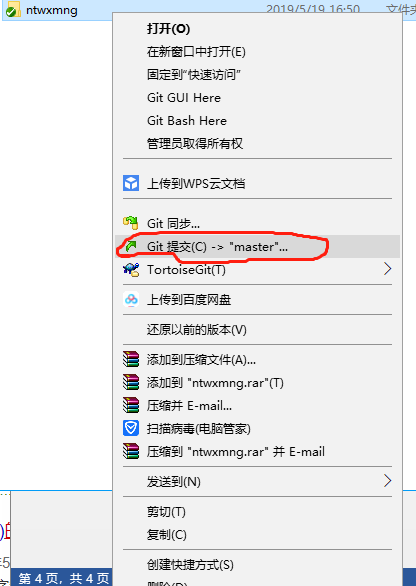
（2）通过拉取（pull）的方式获取最新版后看到成功的标志后方为获取成功。





### 提交版本

（1）提交代码时，选择



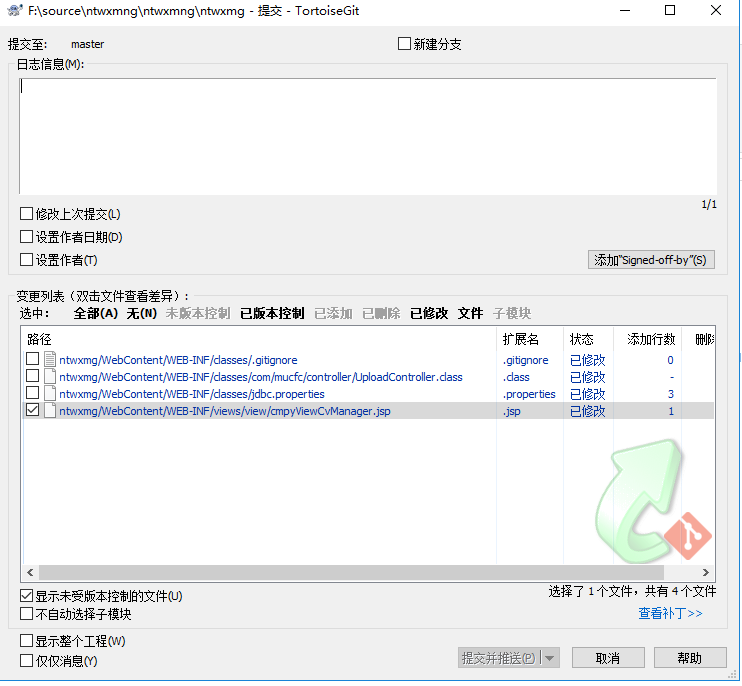
（2）写好注释，并勾选 设置作者日期和设置作者

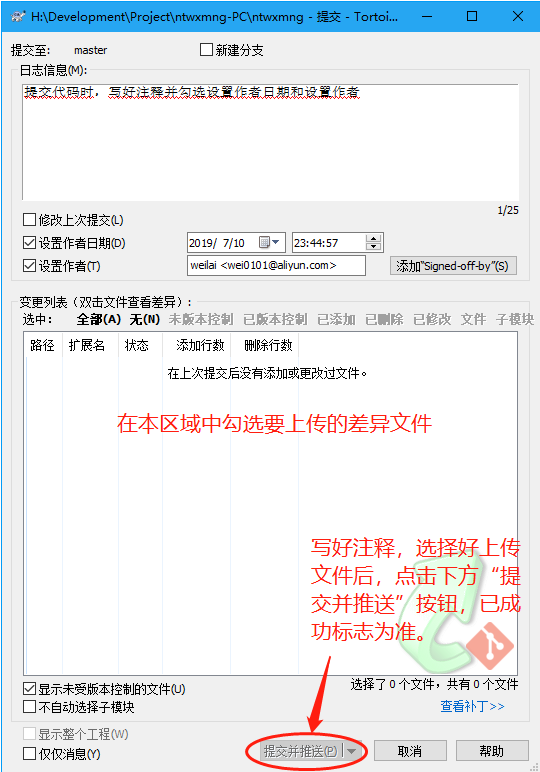
（3）在变更列表中，勾选要上传的文件

（4）**上传时一定要选择 提交并推送。**

如果只选择提交，实际上相当于没有上传完，只有**推送**成功了才算版本上传成功。推送相当于commit操作。另外，如果进行多次提交，最后一次才推送，那么会将之前提交的所有记录也都会按提交顺序一并提交（慎用！会有记录）。

提交修改的文件：

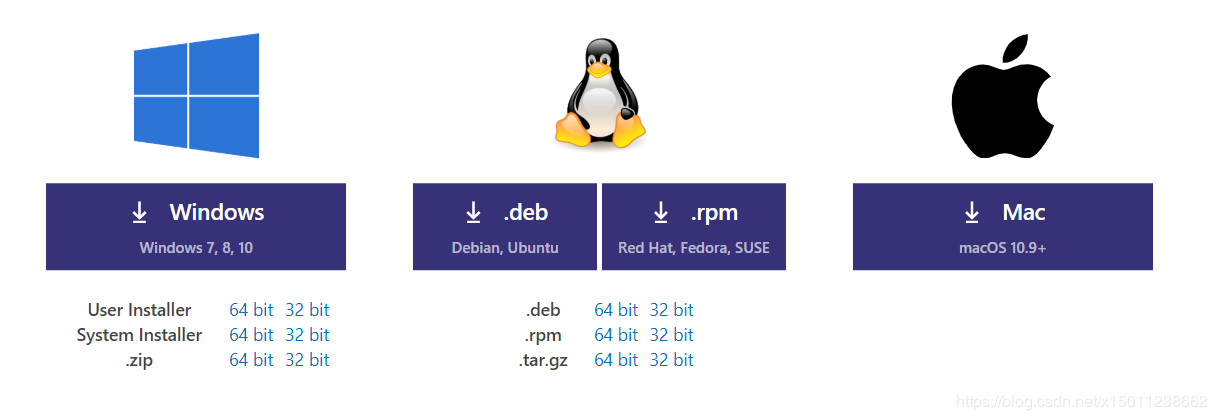




# 开发工具安装

前两个为安装包

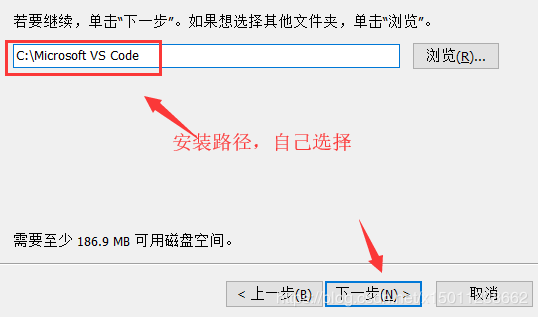
1. 首先下载vscode安装软件 <https://code.visualstudio.com/download>
2. 选择vscode版本（根据自己的操作系统选择相应的版本）

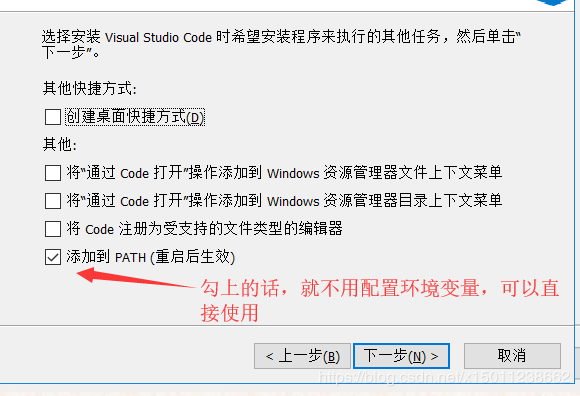


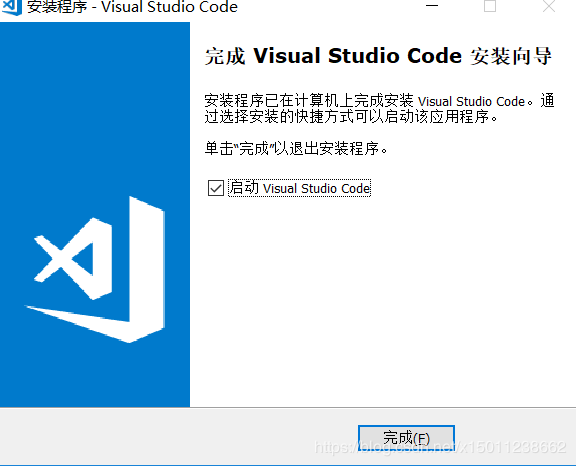
1. 把vscode安装文件准备好：我使用的是64位的操作系统

IMG_256

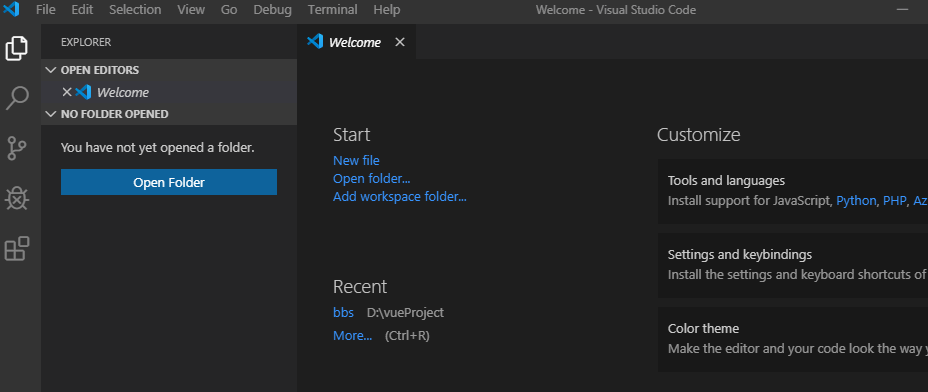
1. 傻瓜式安装，一直下一步，直接到完成



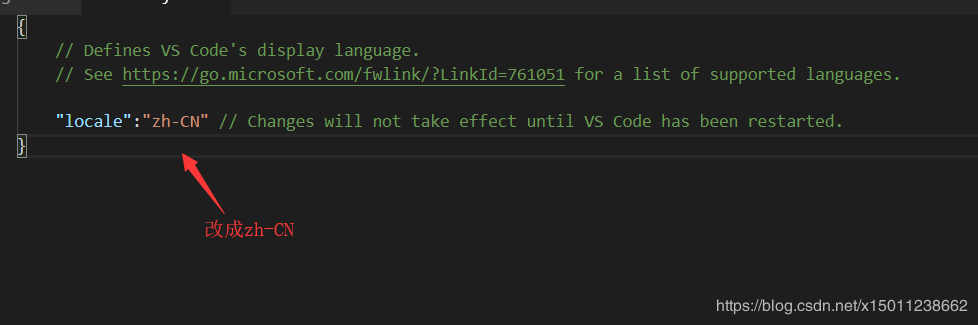
1. 环境变量（这步骤也是默认的，因为改编辑器是微软提供的，会自动配置path）  
   　
2. 安装成功



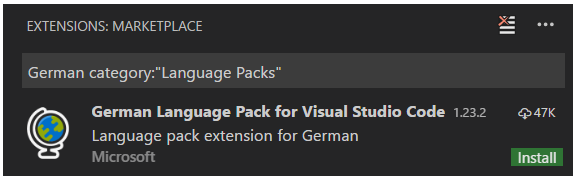
1. 启动



1. 因为刚安装的VSCode默认是英文的，这里需要给设置成中文  
   　　 可以参考官网 <https://code.visualstudio.com/docs/getstarted/locales>
2. 将默认英文改成中文  
   　　 　步骤１.Ctrl+Shift+P打开搜索框输入Configure Display Language命令



修改完成之后，CTRL＋Ｓ保存  
　　 　步骤２：要查找的语言在“扩展”视图（Ctrl + Shift + X）中搜索语言包



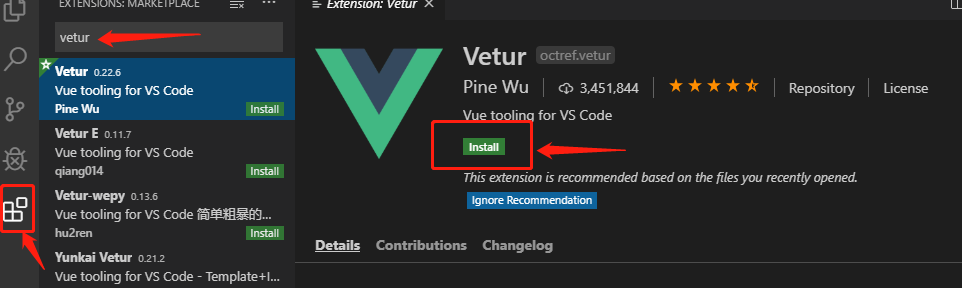
步骤３：重启就可以了，重启之后，就变成中文版了！！！

# 使用vscode创建vue项目

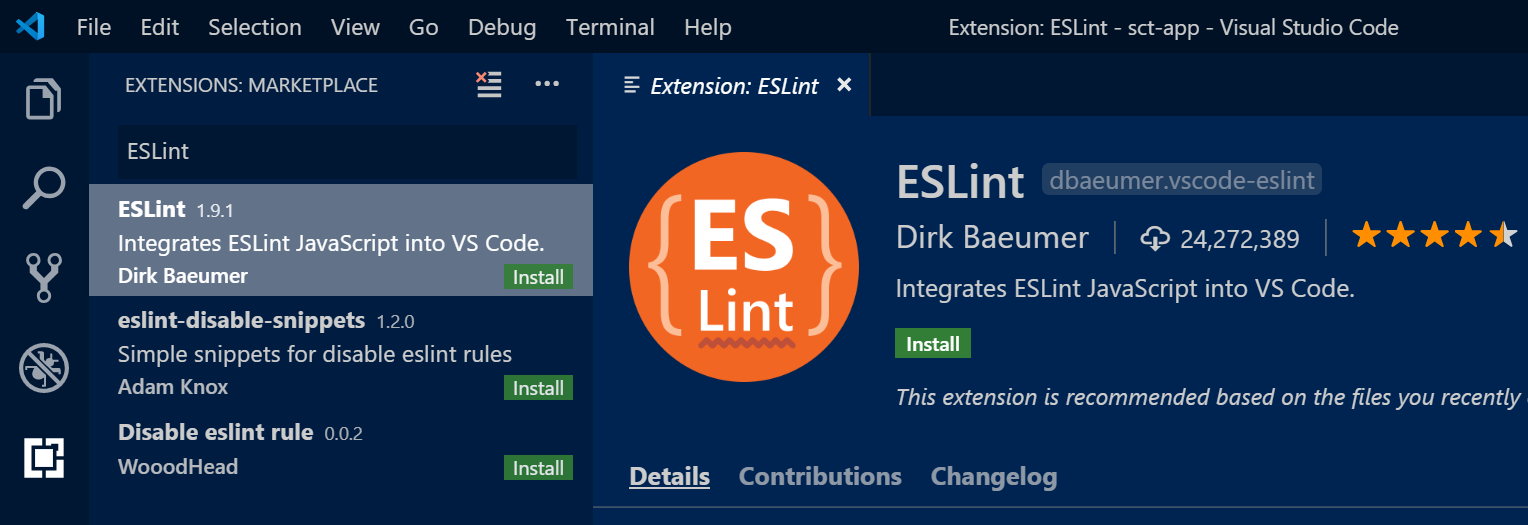
## 安装插件

安装vue插件vetur，实现支持vue文件的代码高亮

安装插件： 点击左边的Extensions图标，输入 vetur ，找对对应版本然后点击install即可



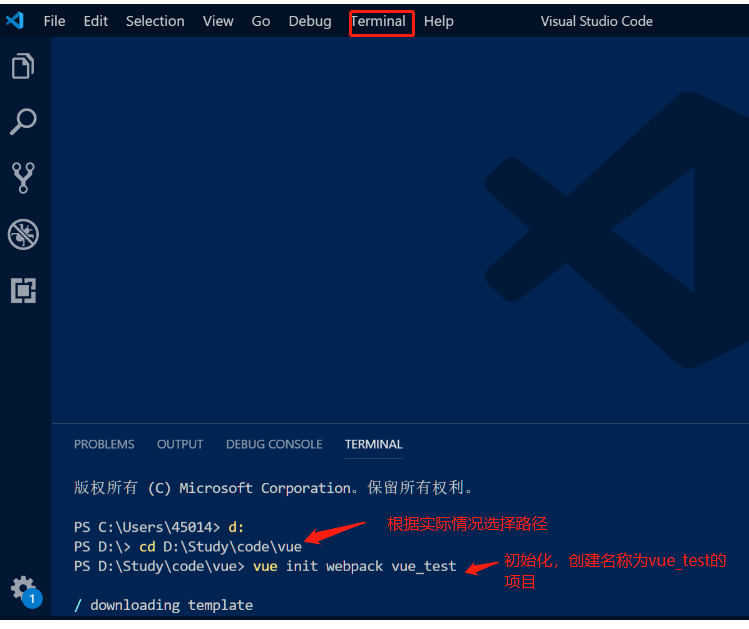
同样方式安装ESLint 插件（代码格式校验）：



## 使用脚手架工具安装项目

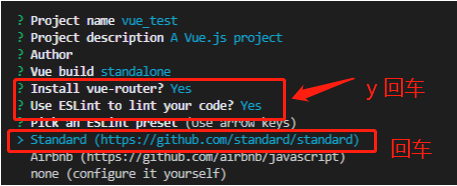
点击Terminal菜单，选择new Terminal，创建一个命令行终端窗口，然后切换到对应的准备存放代码的目录

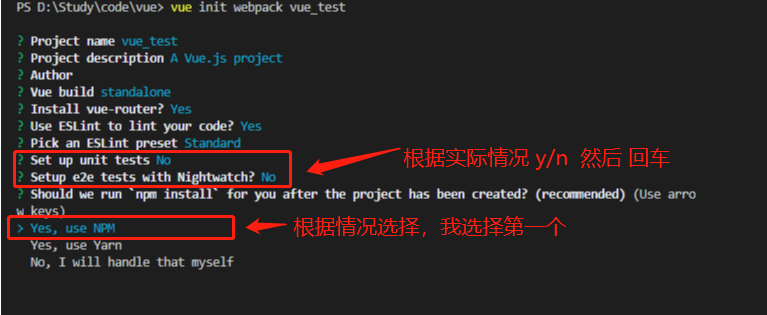
输入：**vue init webpack vue\_tes**t回车，然后输入工程名称**vue\_test**

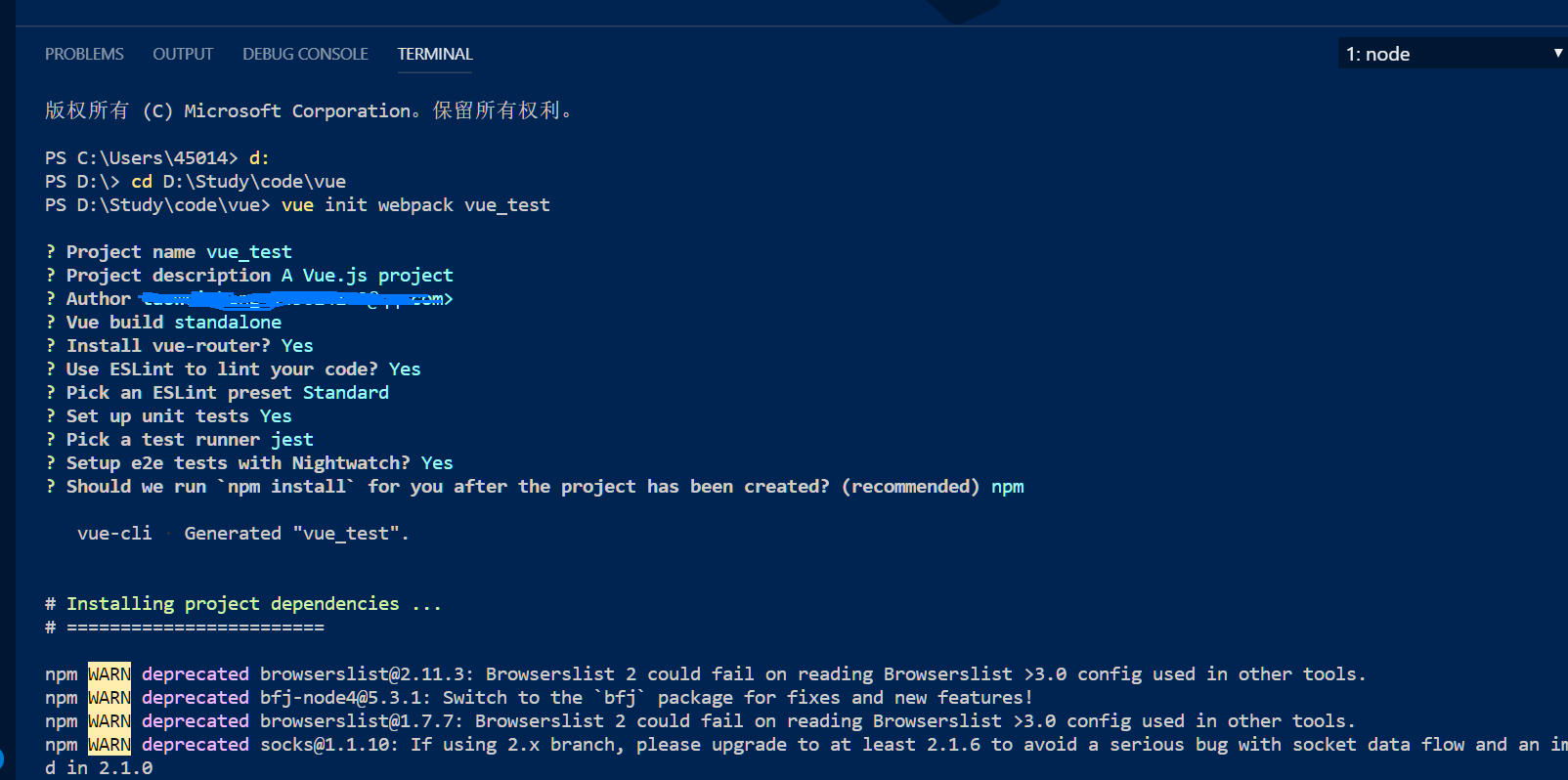


根据自己的情况选择

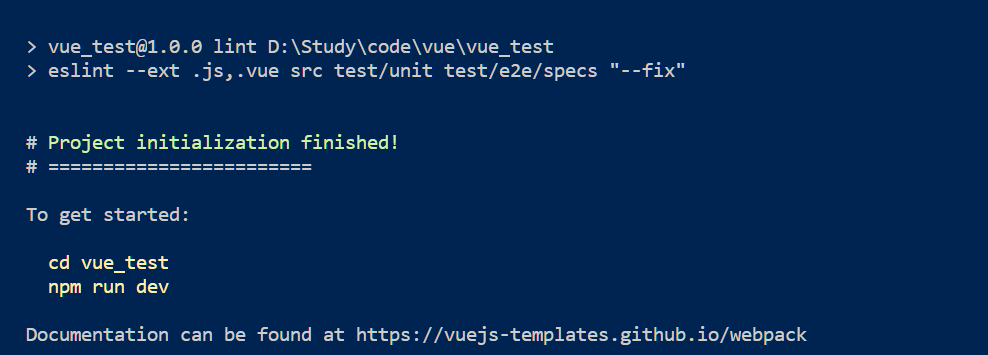








如下提示则成功：



按照说明输入

PS D:\Study\code\vue> cd vue\_test

PS D:\Study\code\vue\vue\_test> npm run dev

> vue\_test@1.0.0 dev D:\Study\code\vue\vue\_test

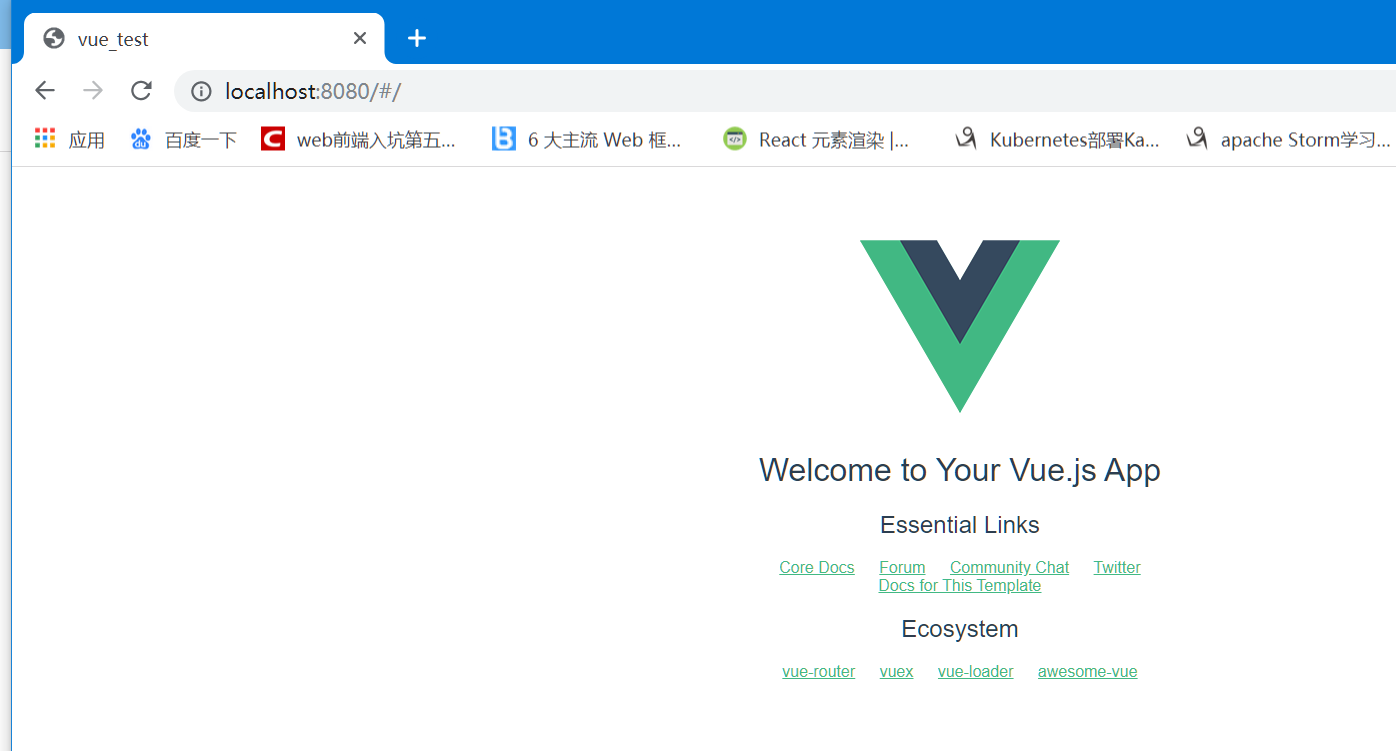
> webpack-dev-server --inline --progress --config build/webpack.dev.conf.js

13% building modules 27/31 modules 4 active ...:\Study\code\vue\vue\_test\src 95% emitting

DONE Compiled successfully in 6043ms 9:10:00 PM

I Your application is running here: http://localhost:8080

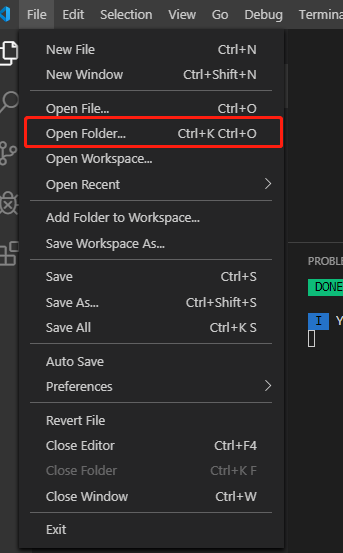
打开浏览器输入：<http://localhost:8080>



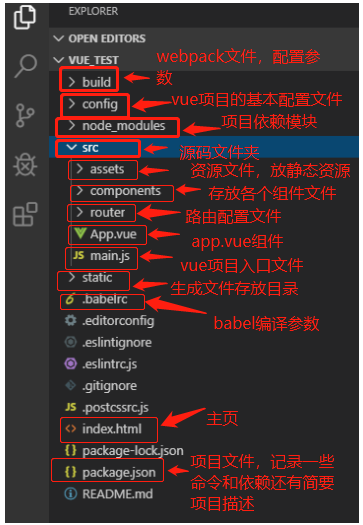
## 项目结构

1.导入项目

使用vscode导入项目file--》open folder---》选择新项目路径 d:/Study/code/vue/vue\_test



1. 项目结构介绍



## Git仓库下载代码

1从远程仓库clone代码

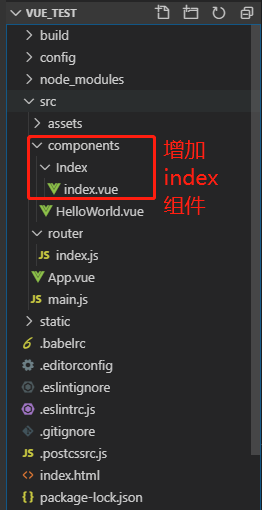
2在本地使用git工具克隆，放到相应的开发目录下

3使用vscode打开代码，在终端中执行npm install 安装项目依赖，看到项目中会多出一个node\_modules文件目录；

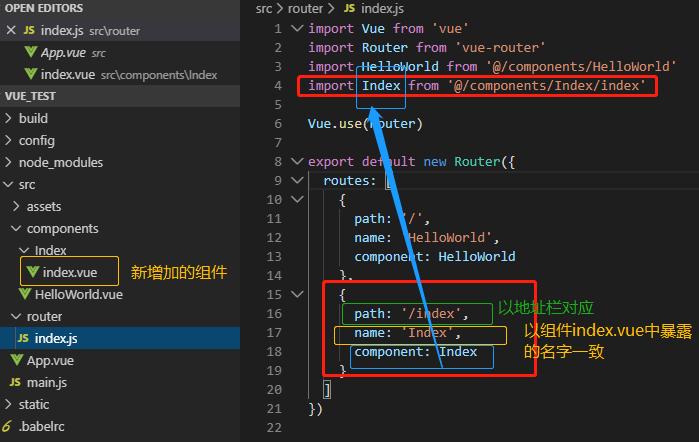
4执行npm run dev启动项目

## 路由使用

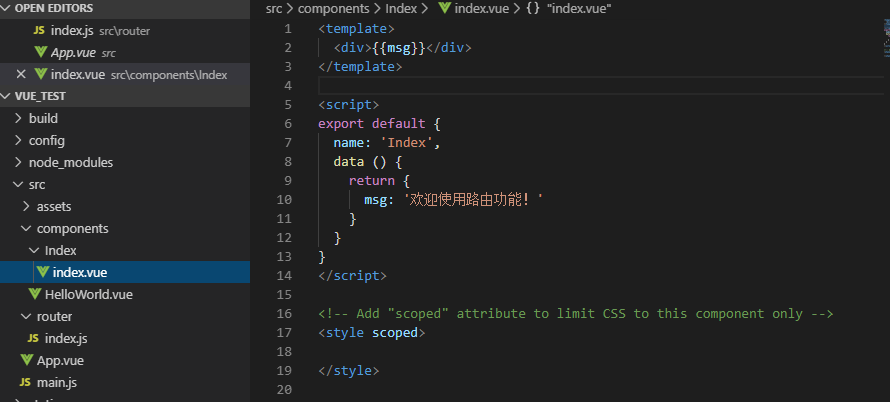
1增加新组建

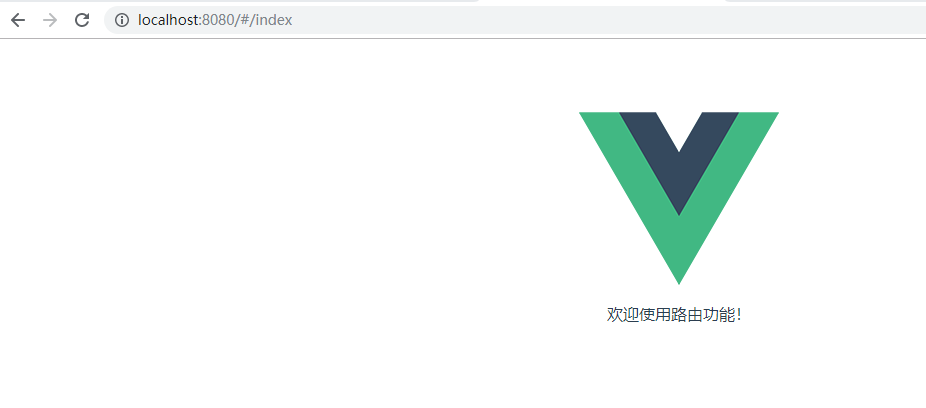


2路由文件中配置



3地址栏访问





去掉地址栏中的#

