技术栈版本：

* node 16
* npm 8.1
* vue-cli 4.0

# 1、创建一个项目

C:\Users\lizhenliang>vue create hello-vue

Vue CLI v4.5.15

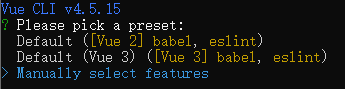
? Please pick a preset: (Use arrow keys)

> Default ([Vue 2] babel, eslint) # vue2版本，默认安装babel、eslint依赖

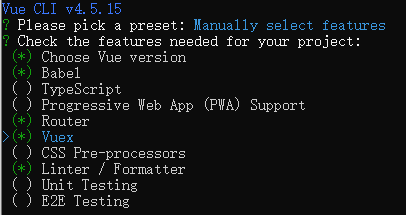
Default (Vue 3) ([Vue 3] babel, eslint) # vue3版本

Manually select features # 手动配置

上下键选择。



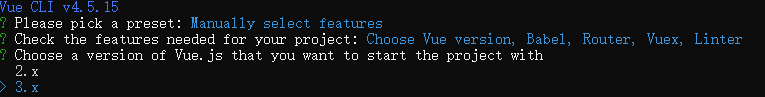
* vue2版本，默认安装babel、eslint依赖
* vue3版本
* 手动配置



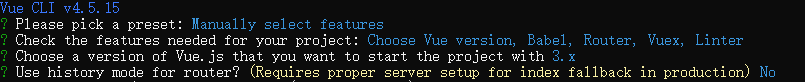
上下键+空格键选择

* 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
* Progressive Web App (PWA) Support：渐进式Web应用程序
* Router：vue-router（vue路由）
* Vuex：vue的状态管理模式，先不选择
* 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Vue版本：**



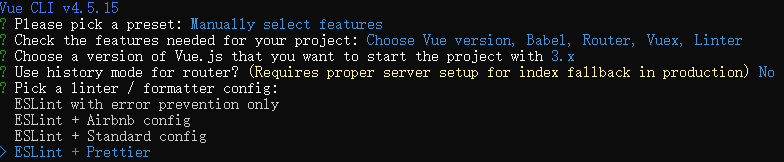
# 2、选择是否使用history router



Vue-Router 利用了浏览器自身的hash 模式和 history 模式的特性来实现前端路由（通过调用浏览器提供的接口）。

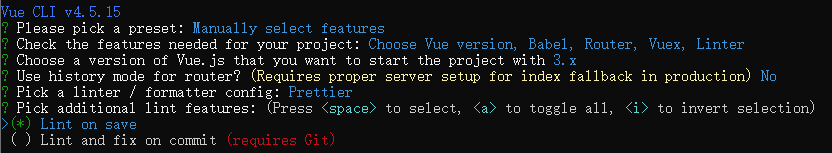
我这里建议选n。这样打包出来丢到服务器上可以直接使用了，后期要用的话，也可以自己再开起来。选yes的话需要服务器那边再进行设置。

# 3、选择ESLint代码效验规则



提供一个插件化的javascript代码检测工具，ESLint + Prettier+使用较多

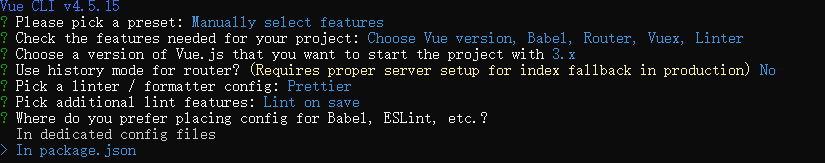
# 4、选择什么时候进行代码效验



* Lint on save： 保存就检查
* Lint and fix on commit： fix 或者 commit 的时候检查

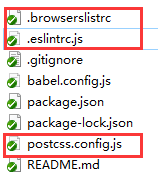
建议选择保存就检测，等到commit的时候，问题可能都已经积累很多了。

# 5、选择如何保存配置

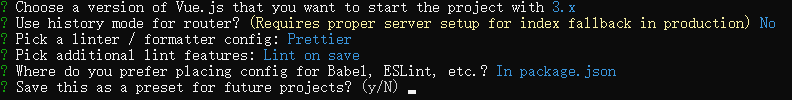


* In dedicated config files： 存放到独立文件中
* In package.json： 存放到 package.json 中  
  这里选择In package.json。

如果选择独立文件，这里会多几个文件：

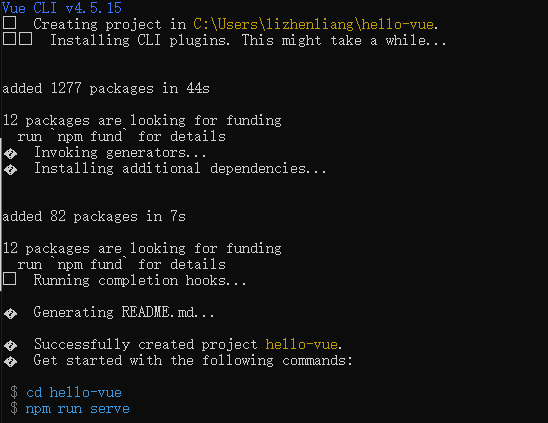


# 6、选择是否保存本次配置



N 不记录，如果选择 Y 需要输入保存名字，下次搭建项目时可以直接选择此次配置。

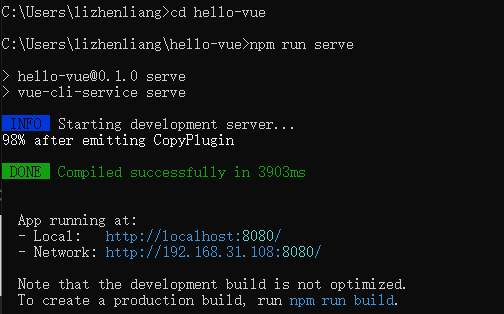
# 7、启动项目



启动项目：

C:\Users\lizhenliang>cd hello-vue

C:\Users\lizhenliang\hello-vue>npm run serve



访问地址：

