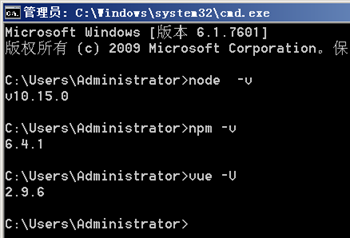
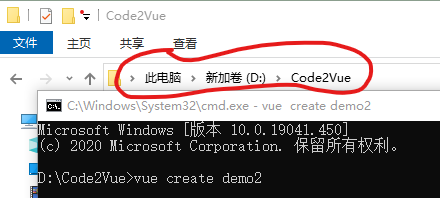
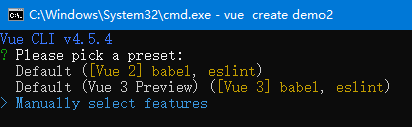
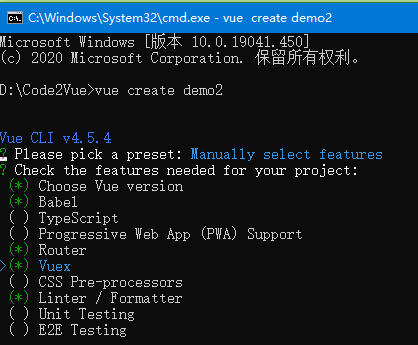
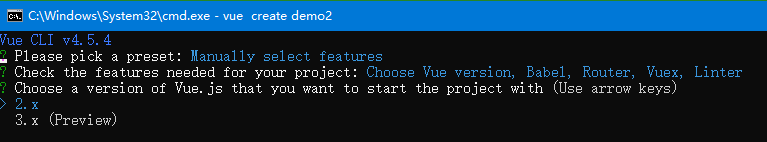
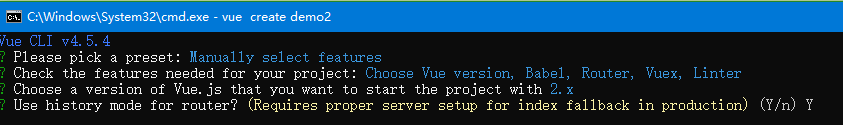
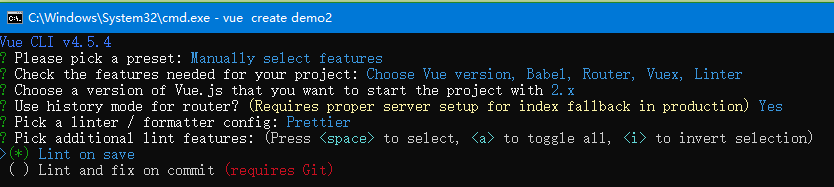
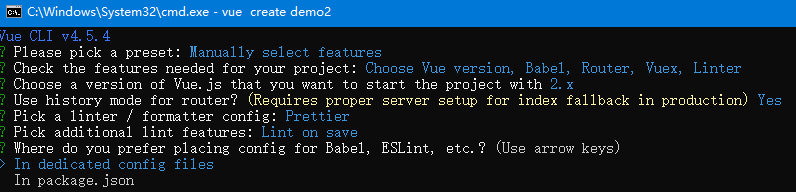
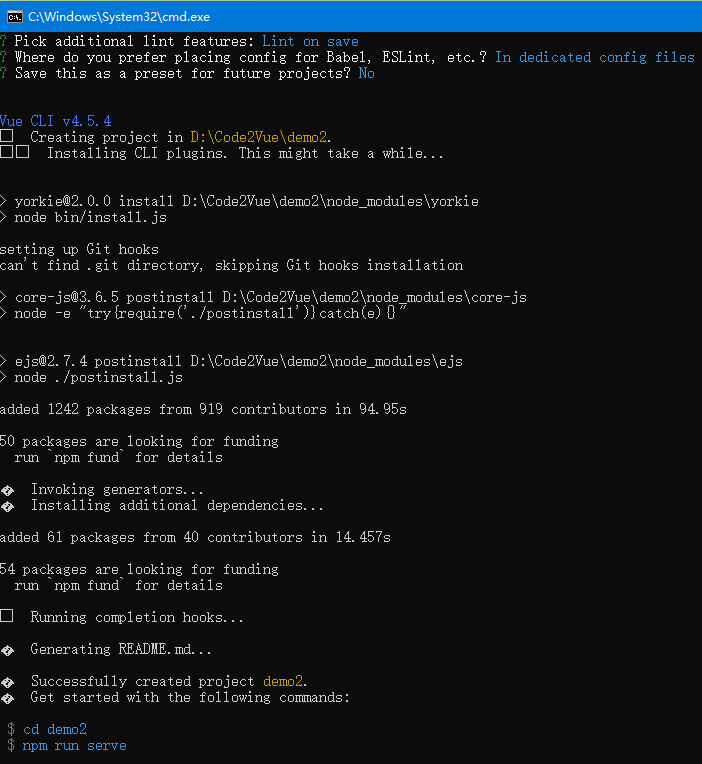
**vue**

*安装卸载脚手架全局安装*

1. 先安装node环境配置 <http://nodejs.cn/> 在node环境下全局安装 cnpm npm install -g cnpm --registry=<https://registry.npm.taobao.org> yarn 安装 npm install -g yarn
2. node -v npm -v cnpm -v vue -V vue --version 查看当前安装项目版本 
3. cnpm install -g @vue/cli 4.5.4 版本升级也可以覆盖 npm install -g vue-cli 2.9.6 vue -V
4. 
5. 创建项目 vue create ... 项目名 

**Vue 手动创建项目流程**

1. 输入vue creat命令后，显示如下图，可以选择使用系统预先定好的模板还是手动选择模块方式来创建项目。 由于默认模板中选中的模块较少，仅提供 babel 和 eslint 支持，所以我们选择手动添加需要的配置。在下图中使用键盘的**上下键**移动光标到:Manually select features（选中后，选项变蓝色）。然后回车进入下一步
2. 这里我们在默认的模块基础上增加Router、Vuex。用键盘的上下键移动光标，用**空格键来选中**所需要的模块。选择完毕后回车进入下一步。
3. Choose Vue version                 选择Vue版本  
   Babel                               将es6 语法转换成兼容的 js  
   TypeScript                         支持使用 TypeScript 语法来编写代码  
   Progressive Web APP(PWA) Support   PWA 支持  
   Router                             vue 路由配置插件  
   Vuex                               vue 程序状态管理  
   CSS Pre-processors                 css 预处理  
   Linter / Formatter                 代码风格检查和格式化  
   Unit Testing                       单元测试  
   E2E Testing                         E2E 测试
4. 由于Vue 3还没有正式发布，建议非尝鲜项目暂时还是选择 Vue 2.x。
5. 在这里我们选择使用Vue的历史模式，**切记创建项目后，需要对项目进行修改配置**。具体修改方式见： [官方文档](https://router.vuejs.org/zh/guide/essentials/history-mode.html)
6. 一般会选择保存就检测 Lint on save。
7. **选择Babel,PostCSS,ESLint配置存放的位置。**
8. 下次创建项目可直接使用当前配置方式。
9. 当生成如下图最下面的蓝色字体表明项目创建成功
10. npm run serve 启动项目
11. ├── public                     # 静态资源  
    │   ├── favicon.ico               # favicon图标  
    │   └── index.html                 # html模板  
    ├── src                       # 源代码  
    │   ├── components                 # 全局公用组件  
    │   ├── router                     # 路由  
    │   ├── store                     # 全局store管理  
    │   ├── views                     # views所有页面  
    │   ├── App.vue                   # 入口页面  
    │   ├── main.js                   # 入口文件 加载组件 初始化等  
    ├── .borwserslistrc           # 浏览器兼容相关  
    ├── .env.xxx                   # 环境变量配置  
    ├── .eslintrc.js               # eslint 配置项  
    ├── .gitignore                 # git忽略文件设置  
    ├── .babel.config.js           # babel-loader 配置  
    ├── package.json               # package.json  
    └── vue.config.js             # vue-cli 配置

| **序号** | **选项** | **描述** | **选择** |
| --- | --- | --- | --- |
| 1 | Choose Vue version | 选择Vue版本 | Y |
| 2 | Babel | vue项目中普遍使用es6语法，但有时我们的项目需要兼容低版本浏览器，这时就需要引入babel插件，将es6转成es5 | Y |
| 3 | TypeScript | TypeScript通过添加类型来扩展JavaScript。通过了解JavaScript，TypeScript可以节省您捕获错误的时间并在运行代码之前提供修复。任何浏览器，任何操作系统，任何运行JavaScript的地方。 完全开源 |  |
| 4 | Progressive Web App (PWA) Support | 渐进式Web应用程序（PWA）支持 |  |
| 5 | Router | 路由 |  |
| 6 | Vuex | Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化 |  |
| 7 | CSS Pre-processors | CSS预处理器，预处理器：比如要用sass或者cssNext就要按照人家规定的语法形式，就是用人家的语法去编写，然后人家把你编写的代码转成css。 |  |
| 8 | Linter / Formatter | 格式化程序 | Y |
| 9 | Unit Testing | 单元测试 |  |
| 10 | E2E Testing | 端到端（end-to-end） |  |

public  
index.html ---入口html页面  
静态资源--json文件  
src  
assets:资源文件（字体，图标，图片）'  
components:存放公共组件  
router:路由文件  
store:存放vuex状态管理  
views:存放页面  
filters:存放过滤器文件  
directives:存放自定义指令文件夹  
mixins:存放混入方法  
utils:存放一些封装的公共方法(js文件)  
http:存放封装的http接口请求  
App.vue 根组件  
main.js 项目执行的入口js  
.gitignore:git提交时忽略文件  
.babel.config.js babel配置文件  
.eslintrc.js EsLint配置文件（代码质量检查配置）  
package.json node配置依赖文件  
README.MD 项目的说明文档  
vue.config.js 配置webpack环境