

一、安装Vue-cli3

打开Node.js command prompt nodejs控制台

1、Vue-cli3: `npm install -g @vue/cli`

2、调试插件: `npm install -g @vue/cli-service-global`

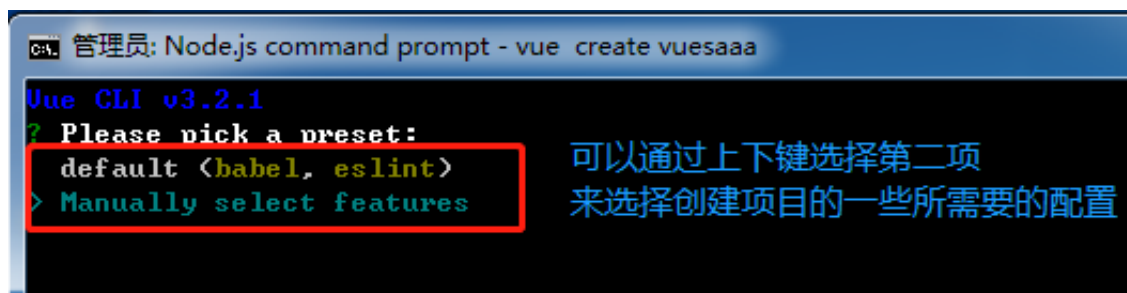
二、新建vue项目，注意：在通过cd命令定位到指定的目录去在执行创建项目

`vue create` 项目名 如: `vue create vue-buy`

创建项目的配置选项:

default (babel, eslint) 默认套餐，提供 [babel](#) 和 [eslint](#) 支持。

Manually select features 自己去选择需要的功能，提供更多的特性选择。比如如果想要支持 TypeScript，就应该选择这一项。



对于每一项的功能，此处做个简单描述： **空格键** 单个选择， **a i** 全选或全不选

TypeScript 支持使用 TypeScript 书写源码。

Progressive Web App (PWA) Support [PWA](#) 支持。

Router 支持 [vue-router](#)。Vuex 支持 [vuex](#)。

CSS Pre-processors 支持 CSS 预处理器。

Linter / Formatter 支持代码风格检查和格式化。

Unit Testing 支持单元测试。E2E Testing 支持 E2E 测试。

```
Vue CLI v3.0.0-beta.6
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select,
<a> to toggle all, <i> to invert selection)
> ☐ TypeScript
☐ Progressive Web App (PWA) Support
☐ Router
☐ Vuex
☐ CSS Pre-processors
☐ Linter / Formatter
☐ Unit Testing
☐ E2E Testing
```

这一步就是要选择配置文件的位置了。对于 Babel、PostCSS 等，都可以有自己的配置文件：.babelrc、.postcssrc 等等，同时也可以把配置信息放在 package.json 里面。

```
Use arrow keys)
> In dedicated config files
  In package.json
```

这个就是问要不要把当前的这一系列选项配置保存起来，方便下一次创建项目时复用 最后询问是否保存，输入n 不保存

```
e.json
? Save this as a preset for future projects? (y/N) n
```

用开发工具打开项目： `npm run serve`

执行： `npm run serve` 启动项目

修改启动命令：

找到package.json文件，打开文件找到 `"serve": "vue-cli-service serve"` 这一行，把前面的 `serve` 修改 `dev` 后保存文件。

```
package.json x
1 {
2   "name": "vue-examination",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "serve": "vue-cli-service serve",
7     "build": "vue-cli-service build",
8     "dev": "vue-cli-service serve",
9     "lint": "vue-cli-service lint"
10  },
11  "dependencies": {
```

把serve 改成dev就可以
用npm run dev来启动了

//改启动命令

改端口：

Vue 3.x中修改端口号则需要在项目根目录下创建一个vue.config.js，内容如下。

vue3.0x版本的

```
module.exports = {
  devServer: {
    port: 8080,    // 端口号
  }
};
```

打包上线 npm run build

在开发完项目之后，就应该打包上线了。

vue-cli 也提供了打包的命令，在项目根目录下执行：npm run build执行完之后，可以看到在项目根目录下多出了一个 dist 目录，该目录下就是打包好的所有静态资源，直接部署到静态资源服务器就好了。