

一、安装步骤

```
Ammy@AmmyComputer MINGW64 /f/2018/project
$ vue init webpack vueProject

? Project name (vueProject) vue-project ①
? Project name vue-project
? Project description (A Vue.js project) ②
? Project description A Vue.js project
? Author (ammy <ammy0620@aliyun.com>) ③
? Author ammy <ammy0620@aliyun.com>
? Vue build standalone
? Install vue-router? (Y/n) y ④
? Install vue-router? Yes
? Use ESLint to lint your code? (Y/n) y ⑤
? Use ESLint to lint your code? Yes
? Pick an ESLint preset (Use arrow keys)
? Pick an ESLint preset Standard
? Set up unit tests (Y/n) y ⑥
? Set up unit tests Yes
? Pick a test runner (Use arrow keys)
? Pick a test runner jest
? Setup e2e tests with Nightwatch? (Y/n) y ⑦
? Setup e2e tests with Nightwatch? Yes
? Should we run `npm install` for you after the project has been created? (recom
? Should we run `npm install` for you after the project has been created? (recom
mended) npm
```

1. Project name :项目名称， 如果不需要更改直接回车就可以了。注意：这里不能使用大写。
2. Project description:项目描述， 默认为A Vue.js project,直接回车， 不用编写。
3. Author： 作者， 如果你有配置git， 他会读取.ssh文件中的user。
4. vue build standalone 直接回车， 建立官方模板；
5. Install vue-router? 是否安装vue的路由插件， Y代表安装， N无需安装， 下面的命令也是一样的。
6. Use ESLint to lint your code? 是否用ESLint来限制你的代码错误和风格
7. setup unit tests with Karma + Mocha? 是否需要安装单元测试工具 Karma+Mocha。
8. Setup e2e tests with Nightwatch?是否安装e2e来进行用户行为模拟测试。
9. Should we run npm install for you after the project has been created? (recommended)npm

二、目录解析

本文主要分析开发（dev）和构建（build）两个过程涉及到的文件，故下面文件结构仅列出相应的内容。

```
-- build // 项目构建(webpack)相关代码
| |-- build is // 生产环境构建代码
| |-- check-version is // 检查node、npm等版本
| |-- utils is // 构建工具相关
| |-- vue-loader.conf is // webpack loader配置
| |-- webpack.base.conf.js // webpack基础配置
```

```

|-- webpack.dev.conf.js // webpack开发环境配置 构建开发本地服务器
|-- webpack.prod.conf.js // webpack生产环境配置
|-- config // 项目开发环境配置
|-- dev.env.js // 开发环境变量
|-- index.js // 项目一些配置变量
|-- prod.env.js // 生产环境变量
|-- test.env.js // 测试脚本的配置
|-- src // 源码目录
|-- components // vue所有组件
|-- router // vue的路由管理
|-- App.vue // 页面入口文件
|-- main.js // 程序入口文件，加载各种公共组件
|-- static // 静态文件，比如一些图片，json数据等
|-- test // 测试文件
|-- e2e // e2e 测试
|-- unit // 单元测试
|-- babelrc // ES6语法编译配置
|-- editorconfig // 定义代码格式
|-- eslintignore // eslint检测代码忽略的文件（来）
|-- eslintrc.js // 定义eslint的plugins,extends,rules
|-- gitignore // git上传需要忽略的文件格式
|-- postcssrc // postcss配置文件
|-- README.md // 项目说明，markdown文档
|-- index.html // 访问的页面
|-- package.json // 项目基本信息,包依赖信息等

```

三、项目命名

运行：npm run dev

(实际是执行的webpack-dev-server --inline --progress --config

build/webpack.dev.conf.js)

打包：npm run build

注意：打包完成后，会生成 dist 文件夹，如果已经修改了文件路径，可以直接打开本地文件查看。

项目上线时，只需要将 dist 文件夹放到服务器就行了。

增加依赖包：

1.npm 模块名 --save-dev

2.--save-dev

自动把模块和版本号添加到模块配置文件package.json中的依赖里devdependencies部分

3. --save-dev 与 --save 的区别

--save 安装包信息将加入到dependencies（生产阶段的依赖）

--save-dev 安装包信息将加入到devDependencies（开发阶段的依赖），所以开发阶段一般使用它



