

vue-cli 脚手架流程

3、安装@vue/cli (Vue CLI 3的包名称由 vue-cli 改成了 @vue/cli)

```
cnpm install -g @vue/cli //yarn global add @vue/cli
```

vue3.x创建项目

```
$ vue create my-pro
```

1.安装vue-cli:

- ① 使用npm (需要安装node环境) 全局安装webpack, 打开命令行工具输入: `npm install webpack -g`或者 (`npm install -g webpack`), 安装完成之后输入 `webpack -v`, 如下图, 如果出现相应的版本号, 则说明安装成功。

注意: webpack 4.X 开始, 需要安装 webpack-cli 依赖, 所以使用这条命令 `npm install webpack webpack-cli -g`

- ② 全局安装vue-cli, 在cmd中输入命令:
 - `npm install -global vue-cli`
 - 安装完成之后输入 `vue -V` (大写的“V”), 如果出现相应的版本号, 说明安装成功。
- 打开C:\Users\Andminster\AppData\Roaming\npm目录下可以看到:
- 打开node_modules也可以看到:

2.用vue-cli来构建项目

① 我首先在D盘新建一个文件夹 (dxi_vue) 作为项目存放地, 然后使用命令行cd进入到项目目录输入: `vue init webpack baoge`

- baoge是自定义的项目名称, 命令执行之后, 会在当前目录生成一个以该名称命名的项目文件夹。

输入命令后, 会跳出几个选项让你回答:

- Project name (baoge): —项目名称, 直接回车, 按照括号中默认名字 (注意这里的

名字不能有大写字母，如果有会报错Sorry, name can no longer contain capital letters)，阮一峰老师博客为什么文件名要小写，可以参考一下。

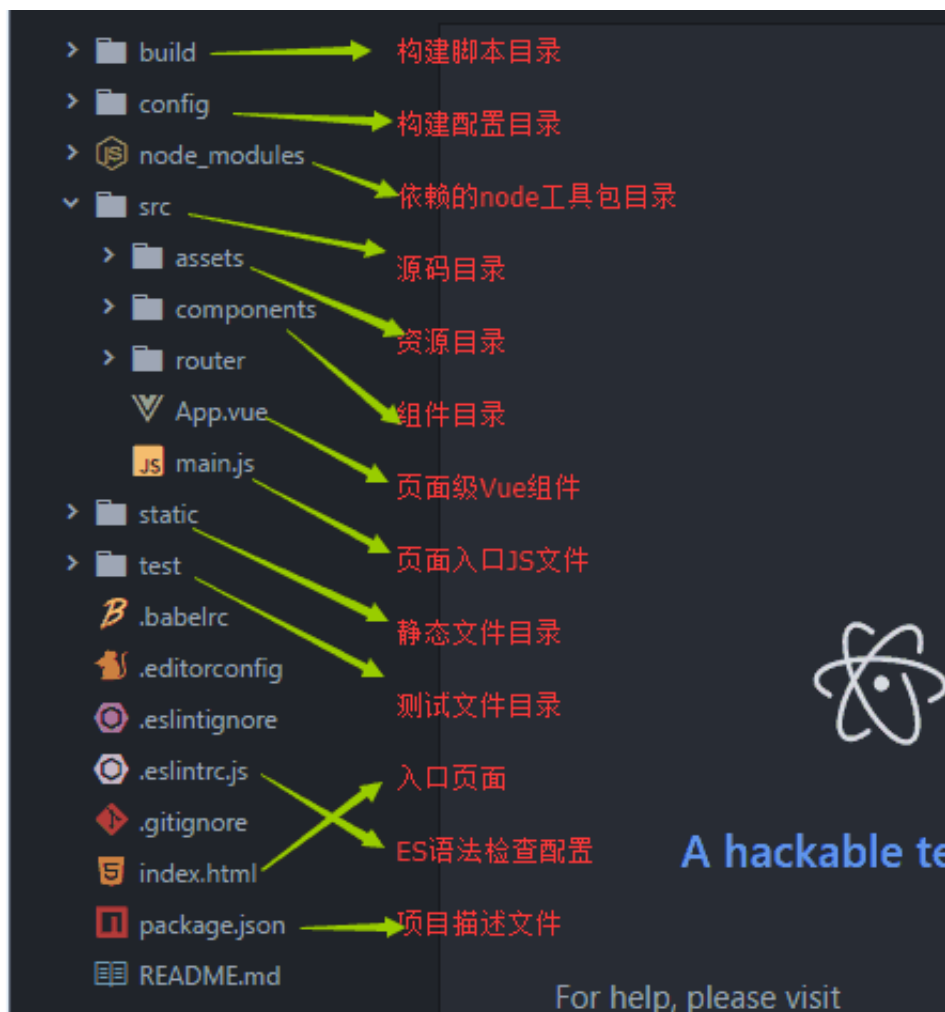
- Project description (A Vue.js project)：—项目描述，也可直接点击回车，使用默认名字
- Author ()：—作者，输入你的大名
接下来会让用户选择：
- Runtime + Compiler: recommended for most users 运行加编译，既然已经说了推荐，就选它了
Runtime-only: about 6KB lighter min+gzip, but templates (or any Vue-specificHTML) are ONLY allowed in .vue files - render functions are required elsewhere 仅运行时，已经有推荐了就选择第一个了
- Install vue-router? (Y/n) 是否安装vue-router，这是官方的路由，大多数情况下都使用，这里就输入“y”后回车即可。
- 接下来也是选择题Pick an ESLint preset (Use arrow keys) 选择一个ESLint预设，编写vue项目时的代码风格，直接y回车
- Setup unit tests with Karma + Mocha? (Y/n) 是否安装单元测试，我选择安装y回车
- Setup e2e tests with Nightwatch(Y/n)? 是否安装e2e测试，我选择安装y回车

回答完毕后上图就开始构建项目了。

② 配置完成后，可以看到目录下多出了一个项目文件夹baoge，然后cd进入这个文件夹：

安装依赖：npm install

每个文件夹代表的意思：



3.启动项目:npm run dev

```

var path = require('path')

module.exports = {
  build: {
    env: require('./prod.env'),
    index: path.resolve(__dirname, '../dist/index.html'),
    assetsRoot: path.resolve(__dirname, '../dist'),
    assetsSubDirectory: 'static',
    assetsPublicPath: './',
    productionSourceMap: true,
    productionGzip: false,
    productionGzipExtensions: ['js', 'css']
  },
  dev: {
    env: require('./dev.env'),
    port: 8075,
    assetsSubDirectory: 'static',
    assetsPublicPath: '/',
    proxyTable: {},
    cssSourceMap: false
  }
}

```

打包后的文件路径

修改端口号

本地调试项目时，建议build 里assetsPublicPath的路径前缀修改为 './'（开始是 '/'），因为打包后，外部引入 js 和 css 文件时，如果路径以 '/' 开头，在本地无法找到对应文件（服务器上没问题）。所以如果需要在本地打开打包后的文件，就修改文件路径。

4.vue-cli的webpack配置分析

```

"scripts": {
  "dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",
  "start": "npm run dev",
  "unit": "jest --config test/unit/jest.conf.js --coverage",
  "e2e": "node test/e2e/runner.js",
  "test": "npm run unit && npm run e2e",
  "lint": "eslint --ext .js,.vue src test/unit test/e2e/specs",
  "build": "node build/build.js"
},

```

- dev中的设置，build/webpack.dev.conf.js，该文件是开发环境中webpack的配置入口。
- 在webpack.dev.conf.js中出现webpack.base.conf.js，这个文件是开发环境和生产环境，甚至测试环境，这些环境的公共webpack配置。这个文件相当重要。
- 还有config/index.js、build/utils.js、build/build.js等，具体请看这篇介绍：<https://segmentfault.com/a/1190000008644830>

5.打包上线

- 自己的项目文件都需要放到 src 文件夹下。
- 项目开发完成之后，可以输入 `npm run build` 来进行打包工作。
- 备注：
 - 1.npm 开启了`npm run dev`以后怎么退出或关闭？
`ctrl+c`
 - 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
 - 1. -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

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。