**[vue-cli（vue脚手架）超详细教程](https://www.cnblogs.com/ming1025/p/9887247.html)**

 都说Vue2简单上手容易，的确，看了官方文档确实觉得上手很快，除了ES6语法和webpack的配置让你感到陌生，重要的是思路的变换，以前用jq随便拿全局变量和修改dom的锤子不能用了，vue只用关心数据本身，不用再频繁繁琐的操作dom，注册事件、监听事件、取消事件。。。。（确实很烦）。vue的官方文档还是不错的，由浅到深，如果不使用构建工具确实用的很爽，但是这在实际项目应用中是不可能的，当用vue-cli构建一个工程的时候，发现官方文档还是不够用，需要熟练掌握es6，而vue的全家桶(vue-cli,vue-router,vue-resource,vuex)还是都要上的。

vue-cli这个构建工具大大降低了webpack的使用难度，支持热更新，有webpack-dev-server的支持，相当于启动了一个请求服务器，给你搭建了一个测试环境，只关注开发就OK。

**1.安装vue-cli**

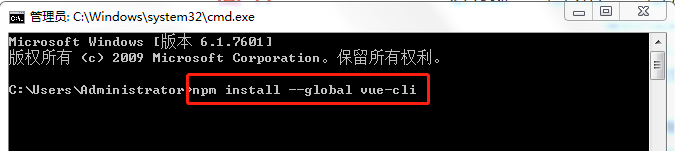
① 使用npm（需要安装node环境）全局安装webpack，打开命令行工具输入：npm install webpack -g或者（npm install -g webpack），安装完成之后输入 webpack -v，如下图，如果出现相应的版本号，则说明安装成功。

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

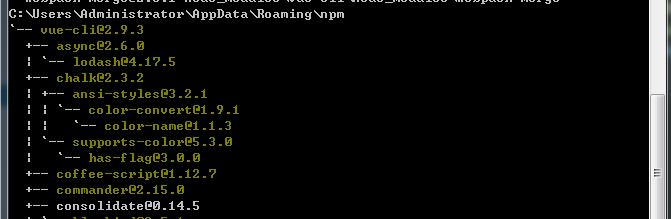


② 全局安装vue-cli，在cmd中输入命令:

（我已经安装过，为了更直观我在电脑上重新演示下）

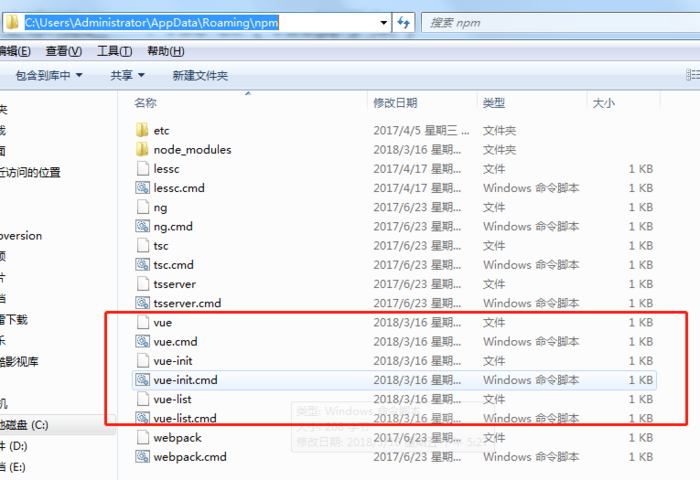


安装成功：



安装完成之后输入 vue -V（注意这里是大写的“V”），如下图，如果出现相应的版本号，则说明安装成功。

打开C:\Users\Andminster\AppData\Roaming\npm目录下可以看到：



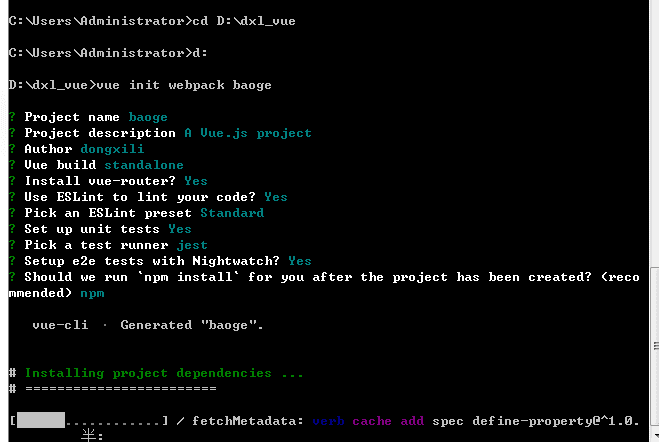
打开node\_modules也可以看到：



**2.用vue-cli来构建项目**

① 我首先在D盘新建一个文件夹（dxl\_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：

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。



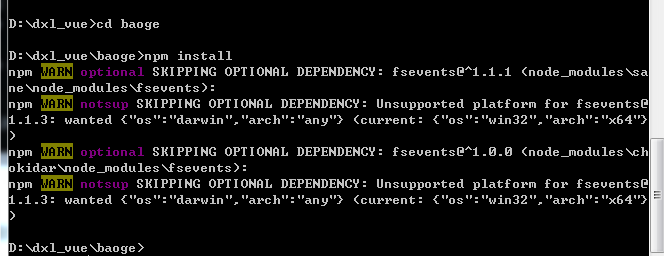
输入命令后，会跳出几个选项让你回答：

* 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](https://link.jianshu.com/?t=http://www.ruanyifeng.com/blog/2017/02/filename-should-be-lowercase.html" \t "_blank) ，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。
* Use ESLint to lint your code? (Y/n) 是否使用ESLint管理代码，ESLint是个代码风格管理工具，是用来统一代码风格的，一般项目中都会使用。  
  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：

 ( 如果安装速度太慢。可以安装淘宝镜像，打开命令行工具，输入：  
 npm install -g cnpm --registry=https://registry.npm.taobao.org  
 然后使用cnpm来安装 )



npm install ：安装所有的模块，如果是安装具体的哪个个模块，在install 后面输入模块的名字即可。而只输入install就会按照项目的根目录下的package.json文件中依赖的模块安装（这个文件里面是不允许有任何注释的），每个使用npm管理的项目都有这个文件，是npm操作的入口文件。因为是初始项目，还没有任何模块，所以我用npm install 安装所有的模块。安装完成后，目录中会多出来一个node\_modules文件夹，这里放的就是所有依赖的模块。

然后现在，baoge文件夹里的目录是这样的：



解释下每个文件夹代表的意思(仔细看一下这张图）：

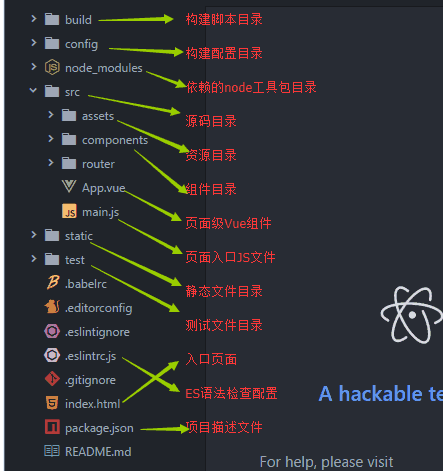


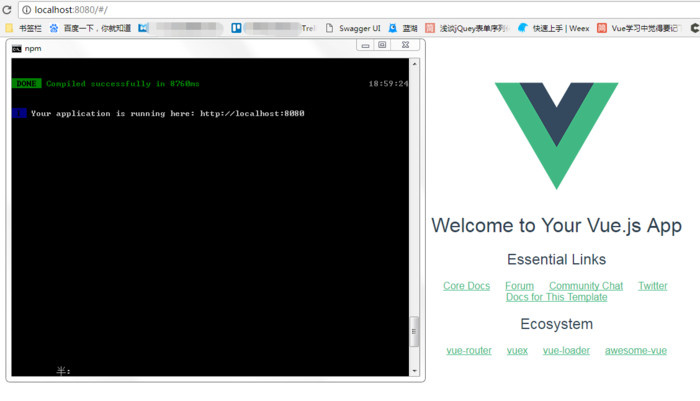
image.png

**3.启动项目**

如果浏览器打开之后，没有加载出页面，有可能是本地的 8080 端口被占用，需要修改一下配置文件 config里的index.js



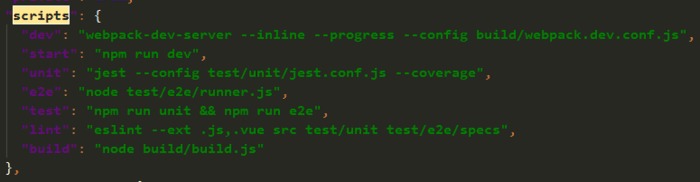
还有，如果本地调试项目时，建议将build 里的assetsPublicPath的路径前缀修改为 ' ./ '（开始是 ' / '），因为打包之后，外部引入 js 和 css 文件时，如果路径以 ' / ' 开头，在本地是无法找到对应文件的（服务器上没问题）。所以如果需要在本地打开打包后的文件，就得修改文件路径。  
我的端口没有被占用，直接成功（服务启动成功后浏览器会默认打开一个“欢迎页面”）：



注意：在进行vue页面调试时，一定要去谷歌商店下载一个vue-tool扩展程序。

**4.vue-cli的webpack配置分析**

* 从package.json可以看到开发和生产环境的入口。



* 可以看到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。可以说，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](https://link.jianshu.com/?t=https%3A%2F%2Fsegmentfault.com%2Fa%2F1190000008644830)

**5.打包上线**

注意，自己的项目文件都需要放到 src 文件夹下。  
在项目开发完成之后，可以输入 npm run build 来进行打包工作。

另：

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。