**完整的Vue脚手架安装**

1. **先安装webpack**

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，会出现相应版本。

② 全局安装vue-cli，在cmd中输入命令:

1. npm install --global vue-cli

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

## 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cmd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。

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：

npm install

( 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。
* 3.启动项目

npm run dev

* 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。

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。

npm run build

另：

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

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

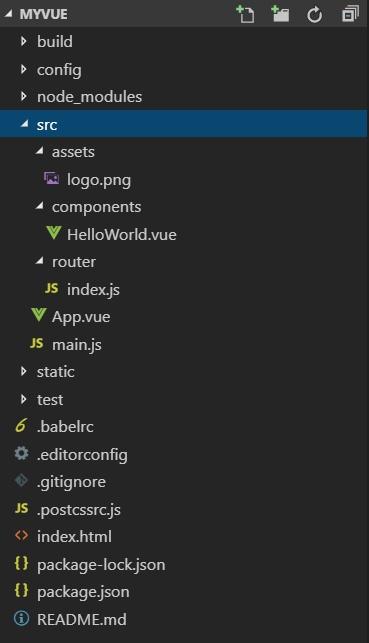
打包完成后，会生成 dist 文件夹，如果已经修改了文件路径，可以直接打开本地文件查看。

项目上线时，只需要将 dist 文件夹放到服务器就行了。

* 打包指令：

cnpm i -g serve  
cd dist  
serve

**vue项目目录说明:**



build：项目构建(webpack)相关代码

config：配置目录，包括端口号等

node\_modules：npm加载的项目依赖块

src：这里是我们要开发的目录，基本上要做的事情都在这个目录里。里面包含了几个目录及文件：

assets: 放置一些图片，如logo等

**components：该目录里存放的我们的开发文件组件，主要的开发文件都存放在这里了**

**App.vue：项目入口文件**

* **main.js：项目的核心文件**
* **router：路由配置目录**

static：放置一些静态资源文件

test：初始测试目录，可删除

.xxxx文件：这些是一些配置文件，包括语法配置，git配置等

**index.html：首页入口文件**

package.json：项目配置文件

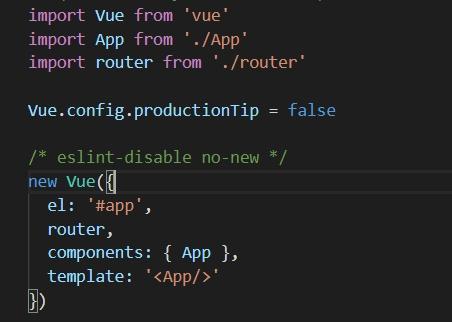
README.md：项目的说明文档，markdown 格式

**三、vue项目启动流程**

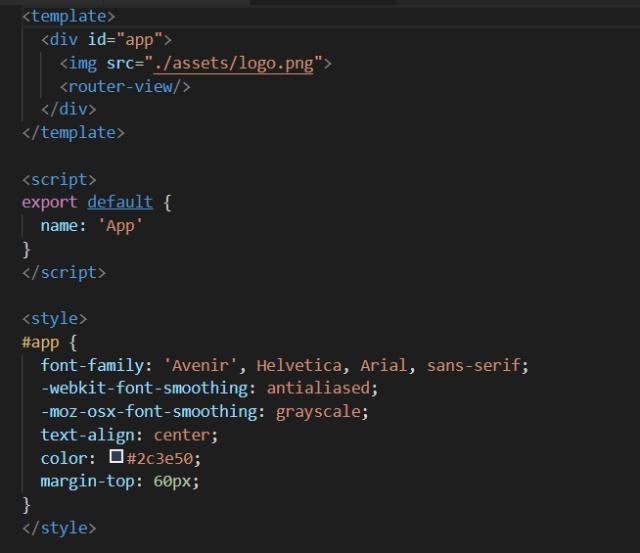
1、在执行npm run dev的时候，会去在当前文件夹下的项目中找package.json文件,启动开发服务器，默认端口是8080；



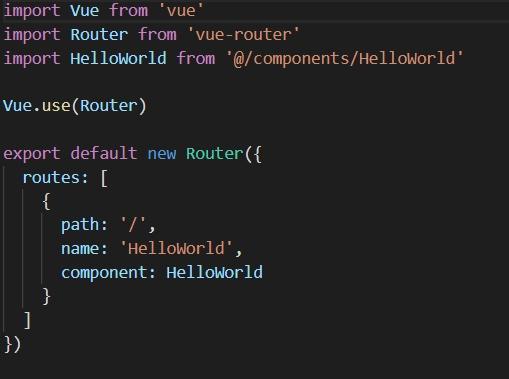
2、找到src的main.js文件，在该文件中new Vue的实例，要加载的模板内容App；



3、App是src目录下的App.vue结尾的文件；



4、在App.vue所对应的模板当中，有一个router-view在src目录下有一个router文件夹，该文件夹有个index.js文件，该文件是配置路由词典，指定了路由地址是空，加载HelloWorld组件

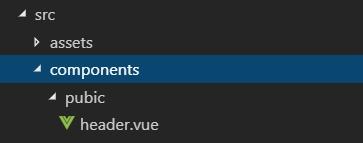


注：vue运行是基于node环境的,构建vue框架之前,需要确保node环境安装成功

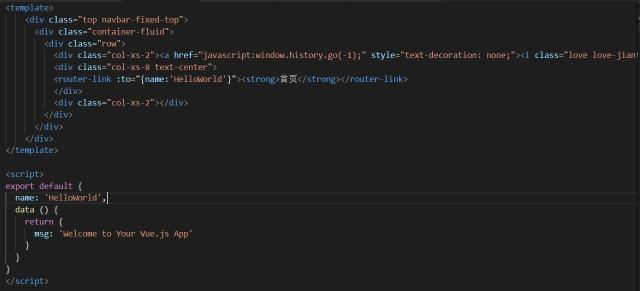
**四、Vue的组件的使用**

1、在components文件夹下创建.vue结尾的文件

例如在：src/components/public/目录下新建header.vue文件



header.vue文件内容如下：



2、在路由配置文件src/router/index.js中引入组件并配置组件路由

2.1、引入组件

import Header from '@/components/public/header'

2.2、配置组件路由

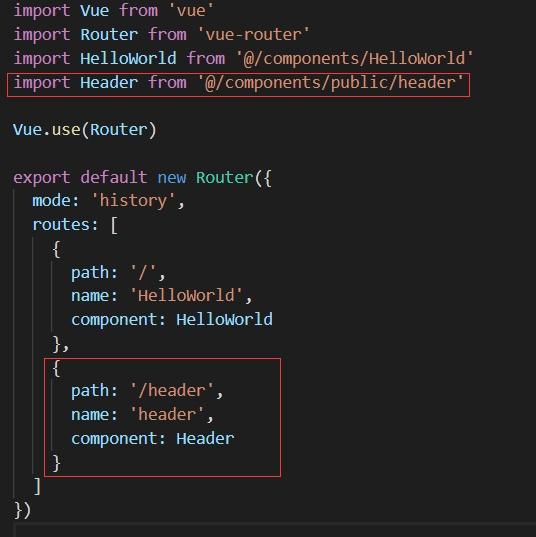
{

path: '/header',

name: 'header',

component: Header

}



3、运行项目：npm run dev

4、在浏览器中输入：localhost:8080/header ,显示如下页面：



**附：vue生命周期示意图**

