**一、使用vue-cli脚手架**

使用vue-cli脚手架要求：

C:\Users\Administrator>node -v

v8.9.3 //要求node版本在6.9以上

C:\Users\Administrator>npm -v

5.5.1 //npm的版本再3.10以上

①安装node.js

②安装：npm install --global vue-clim ;并使用vue -V查看vue版本

③ 使用cd命令：进入项目文件目录C:\xampp\htdocs ，

并使用mkdir vue-app命令创建一个新项目(vue-app目录)，

使用cd命令进入：vue-app新项目目录

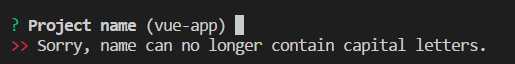
④使用命令vue init webpack vue-app //初始化webpack项目vue-app，其中vue-app为项目文件夹名

//Target directory exists. Continue? Yes 第二次运行vue init webpack vue-app

项目名称中不能有大写



就会出现下边这种情况

 +↑箭头重新输入

Project description vue-cli-test //项目描述

Author jsj //项目作者

> Runtime + Compiler: recommended for most users //选择个 +回车

Runtime-only: about 6KB lighter min+gzip, but templates (or any Vue-specific HTML) are

ONLY allowed in .vue files - render functions are required elsewhere

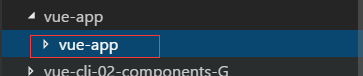
? Install vue-router? No --->安装路由

[? Use ESLint to lint your code? No -----> ESLint是](ESLint的使用.doc)**[一个语法规则和代码风格的检查工具,可以用来保证写出语法正确、风格统一的代码。](ESLint的使用.doc)**

? Set up unit tests No -->设置单元测试

? Setup e2e tests with Nightwatch? No -->用Nightwatch建立E2E测试

? Should we run `npm install` for you after the project has been created? (recommended) (Use arrow keys)

> Yes, use NPM //选这个+回车：就会在vue-app中新增以vue-app项目名称的文件目录

Yes, use Yarn

No, I will handle that myself

? Should we run `npm install` for you after the project has been created? (recommended)

npm vue-cli · Generated "vue-app".

# Installing project dependencies ...

# ========================

[>uglifyjs-webpack-plugin@0.4.6](mailto:>uglifyjs-webpack-plugin@0.4.6) postinstall C:\xampp\htdocs\vue-app\vue-app\node\_modules\webpack\node\_modules\uglifyjs-webpack-plugin

> node lib/post\_install.js

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN ajv-keywords@3.2.0 requires a peer of ajv@^6.0.0 but none is installed. You must install peer dependencies yourself.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.3 (node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1175 packages in 397.335s

# Project initialization finished!

# ========================

To get started:

cd vue-app

npm run dev

Documentation can be found at https://vuejs-templates.github.io/webpackC:\xampp\htdocs >

Yes, use NPM //等这个运行成功后 ，

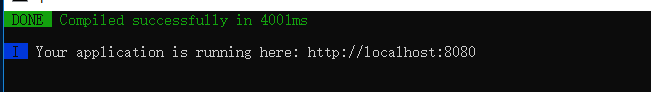


⑤再根据上一步的提示：运行一下命令

5.1-->cd vue-app

5.2-->运行npm install 安装需要的模块

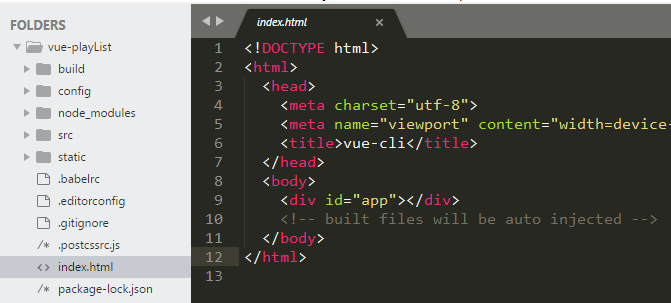
5.3-->npm run dev 运行项目并开启一个8080的一个端口 //或者在浏览器的地址栏中直接输入localhost：8080即可。



⑥将项目文件防止在C:\xampp\htdocs 下；并打开xmapp软件运行MySQL和Apache

| ⑦**项目文件目录解析** | |
| --- | --- |
| **目录/文件** | **说明** |
| build | 最终发布的代码存放位置。 |
| config | 项目的一些配置，包括端口号等。我们初学可以使用默认的。 |
| node\_modules | 项目依赖的代码模块 |
| src | 项目的原码。里面包含了几个目录及文件：  assets: 放置一些图片，如logo等。  components: 目录里面放了一个组件文件，可以不用。  App.vue: 项目的根组件的入口文件，我们也可以直接将组件写这里，而不放在components 目录。  main.js: 是webpack打包项目的一个入口文件  -->项目的核心文件，并在main.js中引入vue模块和App（或App.vue）根组件，并挂载到index.html中  import Vue from 'vue' // ----> 引入vue模块  import App from './App' // ----> 引入App（或App.vue）根组件  new Vue({ //实例化vue  el: '#app', //-- > 挂载到index.html中的#app上  components: { App },// --> 注册App根组件  template: '<App/>' //-- > 使用App模板  }) |
| static | 存放第三方静态资源的目录（如图片、字体等)；如果为空，git就不会将其提交。 |
| .babelrc | 是对babel的一些配置；----比如babel-loader处理ES6语法 |
| .editorconfig | 对代码风格进行配置 --->比如缩进几个字符 |
| .eslintignore | 配置那些文件目录，可以忽略使用[ESLint](ESLint的使用.doc)对语法检查 |
| .eslintrc.js | 配置ESLint |
| .gitignore | 配置git忽略的那些文件目录 |
| index.html | 醒目的入口文件，你可以添加一些 meta 信息或统计代码啥的。 |
| package.json | 项目的配置文件--->一般是项目的一些描述。 |
| [package-lock.json](http://www.sogou.com/link?url=DOb0bgH2eKh1ibpaMGjuy_914-o2bQk_8CYSXPCumgSuSdKpCMIsLny_GD_6DQHAXK4dQ9gQbiA." \t "https://www.sogou.com/_blank) | 是npm install 生成的文件；里面有明确的各依赖模块的版本号，实际安装的结构，也有逻辑树的结构 |
| README.md | 项目的说明文档，markdown 格式 |

**入口容器文件Index.html**



**项目的核心文件main.js**

这里的vue是使用npm install 下载好的引入的模块；；根组件App就是src/APP.vue文件



## 根组件src/APP.vue

<!-- html模板 -->

template是不会渲染再index.html中的

在#app跟标签外和template标签中只能有且只有一个跟标签

<template> <div id="app"> <img src="./assets/logo.png"> <HelloWorld/> </div> </template>

<!--组件逻辑-->

<script>

import HelloWorld from './components/HelloWorld'//引入HelloWorld组件模板 ;

export default { //导出HelloWorld 组件

name: 'app',

components: {

HelloWorld //在这里使用HelloWorld组件模板来，注册HelloWorld局部组件

}

}

</script>

<!-- 组件样式代码 scoped-css的局部作用域 是组件独有的样式 -->

<style scoped> #app { font-family: 'Avenir', Helvetica, Arial, sans-serif; -webkit-font-smoothing: antialiased; -moz-osx-font-smoothing: grayscale; text-align: center; color: #2c3e50; margin-top: 60px; } </style>

总结组件的使用：

①先有一个XX.vue的组件模板文件；②使用import引入组件模板；③使用components来注册组件；

④用export default抛出组件（APP.vue根组件是直接挂载到index.html页面中的）；⑤使用组件

⑧直接运行做好的项目：cd 命令进入项目--->运行：npm run dev 命令-->打开浏览器127.0.0.1:8080即可

**二、使用路由：**

①若是在新的项目中使用vue-cli脚手架和vue-router时，就要在以上使用vue-cli脚手架的第④步时：

? Install vue-router? Yes --->安装vue-router

① npm install vue-router --save-dev