# 展会项目前台文档

git路径<http://192.168.11.203:3000/department-2/visitor-show-h5.git>

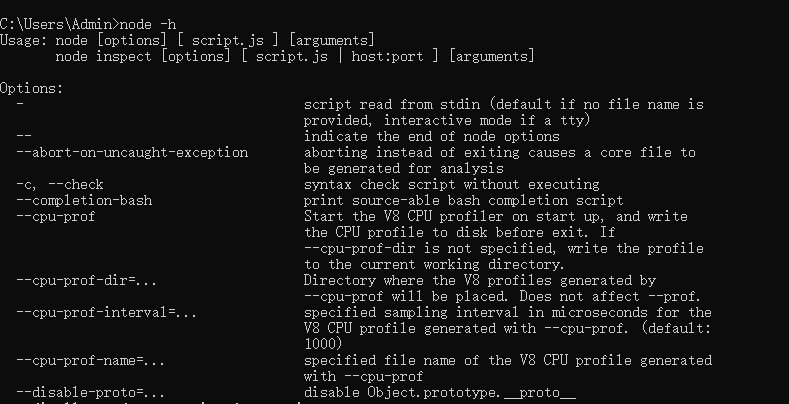
内网环境ajax请求地址 http://192.168.21.90:7001/

线上环境ajax请求地址"http://118.123.247.223:3000/api"

## 开发环境配置及项目部署

#### 1.1安装node

从node官网上安装node 14.4.0以上的版本(<https://nodejs.org/zh-cn/>)，安装完成后打开cmd输入node -h，如若出现以下内容则代表安装成功。

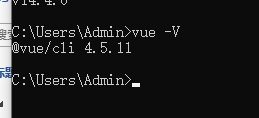


#### 1.2全局安装vue-cli

Vue Cli 是一个基于vue.js进行快速开发的完整系统。

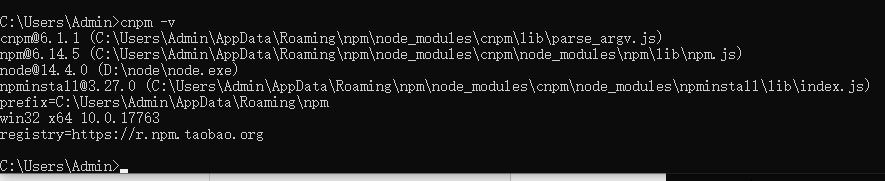
这里所需要安装的版本是4.5.11及以上

打开cmd控制输入命令 npm install –g @vue/cli 来进行全局安装。安装成功后在控制台中输入 vue –V，出现以下内功则代表安装成功。



#### 1.3安装cnpm

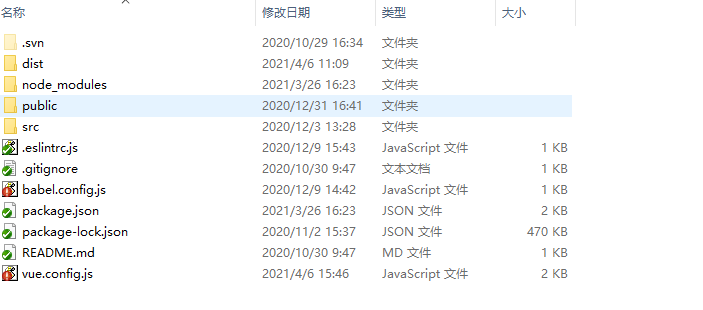
因为前端开发需要npm资源包管理，npm是node中自带的内容，安装完node后，npm也就默认安装了，但是通过npm下载资源的时候是从国外资源下载，速度较慢，这里采用国内的cnpm来下载资源包，打开cmd控制台，执行命令 npm install -g cnpm 全局安装cnpm，安装成功后，输入cnpm –v如果出现以下内容则代表安装成功。



#### 1.4下载项目

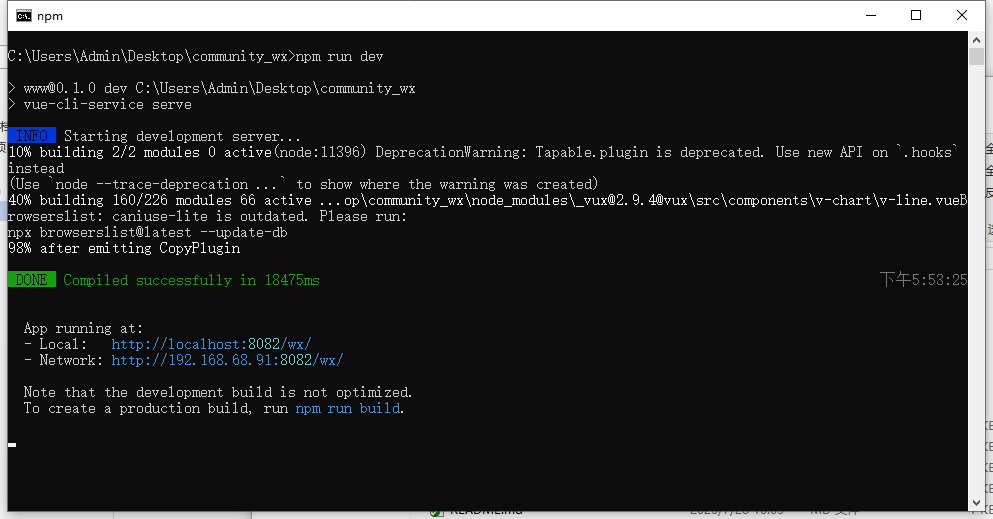
到这里，基本的环境配置已经好了，从svn路径https://192.168.211.43/svn/vmsvn/rxrz-1vsn/trunk/ruoyi-community/web-vue

，导出项目到本地，如图所示



打开cmd控制台，切换到项目的根目录，然后执行cnpm install ，下载项目所需要地资源包（这里资源过大，网速不好可能需要等一会），如若下载完后报错，则重新执行cnpm install,如若多次执行下载命令依然报错，那就检查package.json文件里面的资源包配置是否正确。

下载成功后，在cmd控制台中执行 npm run serve命令，来运行项目，出现如下内容则代表运行成功，通过local或者network的url直接在浏览器中访客即可。



从上述svn路径下载完项目之后，通过cmd打开项目，在项目根目录下执行命令 npm install 或者cnpm install下载所需资源包,项目部署时，运行npm run build命令，将生产的dist文件夹扔到线上服务器所指定目录即可。

#### 1.5注意事项

在使用svn进行项目提交之前，最好先进行更新，以防止发生冲突，在更新和提交的时候，不要选择在项目的根目录下进行，因为在根目录下默认会将node\_modules文件夹也提交上去，这个文件夹是存储项目所需的资源包，是不需要提交到svn服务器上的，如果是要提交或者更新项目的业务逻辑内容，进入src文件夹下进行批量更新和提交操作，如若设计到新的资源包的添加或者对vue-cli配置文件的修改要进行更新或提交，那就在项目的根目录下右键选中要操作的文件，单个文件进行操作。

在用npm或者cnpm下载资源包的时候一定要加上 –D，如我要下载jquery，cnpm i jquery –D，-D将会把所下载的包的信息添加到package.json文件中，只需要提交package.json文件，另一个人在更新的时候，只用更新package.json文件，而不需要更新node\_modules文件夹，更新完这个文件之后，用cnpm install命令，npm将会自动更新package.json文件里对应地资源包。

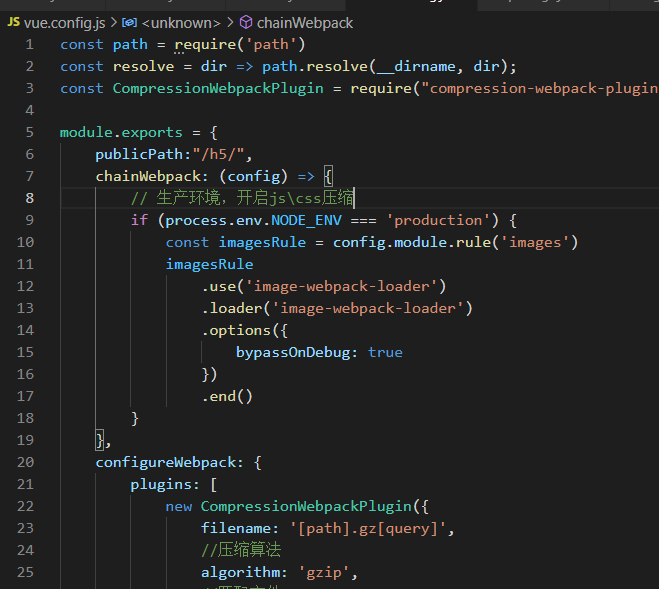
## 项目介绍

### 2.1项目架构介绍

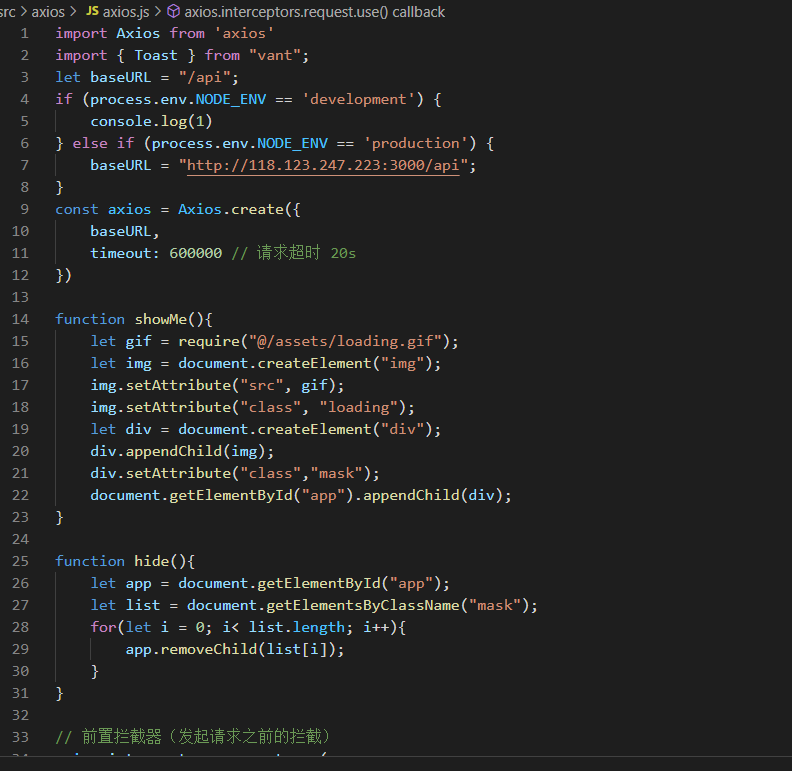
该项目是使用vue2框架来搭建，所使用的核心库有vue-router（用于做网站路由跳转）、axios（用于对ajax请求与响应的封装）、vue-cli脚手架用于快速搭建项目并进行webpack的一些基础配置。

### 2.2项目配置介绍

项目打包的时候，需要在vue.config.js文件中配置 publicPath:’/h5’指定所有静态请求的路径都带上 h5前缀，如下图



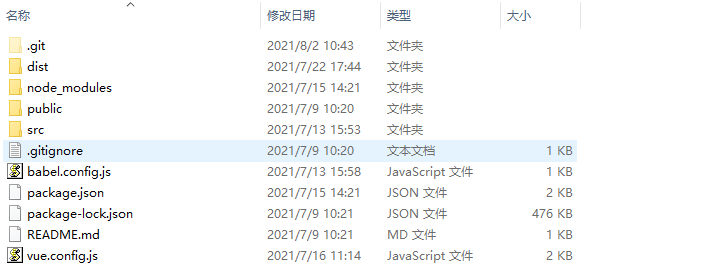
Axios的配置问件是src/axios/axios.js，在这里配置了请求超时时间、请求时的前端loading加载效果、ajax请求前的拦截处理和ajax请求后的拦截处理。



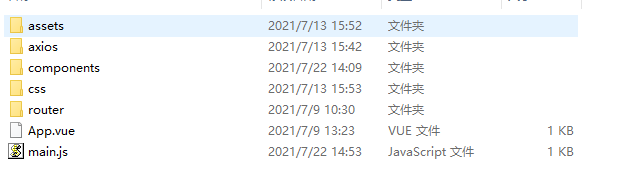
### 2.3项目结构介绍

.git文件夹是git自己生成的文件夹，dist是项目运行npm run build命令后所生成的打包后的项目，用于线上部署，node\_modules是项目所依赖的第三方资源包，也就是通过cnpm install xxx所下载的资源包都在这个文件夹下，（进行项目提交的时候这个文件夹 一定不要提交上去！！！）public文件夹是存放项目html页面的，也是项目唯一的一个html页面，

Package.json文件是项目资源包的配置信息，比如某个资源包的版本号，运行cnpm install命令的时候，其实就是去这个文件下寻找有哪些需要下载的依赖包，vue.config.js是vue-cli的配置文件



下面介绍一下src文件夹，这是项目的核心内容

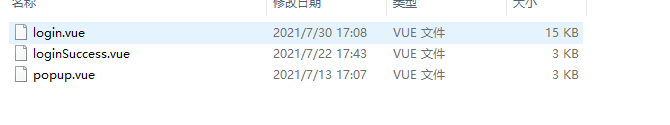


Assets文件夹是用于存放项目的的图片，axios文件夹是用于配置axios库所封装的ajax请求。路由的配置文件在router文件夹下的index.js里面，若果想添加一个路由配置，则打开router文件夹下的index.js文件找到routes数组，在里面新添加一个对象，path属性是路由的uri，name是路由的名称，也可以在路由跳转时使用，meta是该路由的一些元标签，如访问到该路由时，浏览器的title该显示什么，component属性是匹配到该路径时所加载的vue组件，在这里使用import函数来进行组件的懒加载，提升vue首次加载的效率。



Css文件夹下面存放的是所有的css文件。

components文件夹下是用于存放项目组件，所有的vue组件（除App.vue主组件）全部放在该文件夹下，目前有login、loginSuccess、popup三个组件，其中popup组件是个人封装的弹出层组件，用于项目中的查看注册协议功能。



main.js文件是项目的主入口，App.vue是项目第一个加载的vue组件，main.js文件中加载了项目中一些初始化配置文件，比如router、axios配置文件、css样式。

在这里还有一点是进行图片请求的路径是

