# 前端VUE项目安装及运行

1. 克隆项目代码带本地

--git clone http://git.sfc.com/suntekcorps/advtManager.git

1. 安装插件依赖

--进入前端目录：cd /advtManager/Code/Frontend

--安装依赖：npm install

1. 开发环境运行

--本地开发 启动项目：npm run dev（前端目录frontend下运行）

1. 生产环境运行

--打包正式环境：npm run build

1. 开始开发工作
2. 注意事项

--项目运行之前请安装git及nodejs工具，不会的自行百度，nodejs参考：[nodejs正确安装及配置及npm命令无反应的解决方案](http://www.yuxang.com/%e5%85%b3%e4%ba%8enodejs%e7%9a%84npm%e5%91%bd%e4%bb%a4%e6%97%a0%e5%8f%8d%e5%ba%94%e7%9a%84%e8%a7%a3%e5%86%b3%e6%96%b9%e6%a1%88/)。

--本项目技术栈基于 ES2015+、vue、vuex、vue-router 、axios 和 element-ui，所有的请求数据可以通过使用Mock.js模拟，请提前了解和学习这些知识会对使用本项目有很大的帮助。

--项目安装之前请先全局安装vue-cli及webpack工具，建议在本地cmd中运行安装，不建议使用shell工具及gitbash工具安装插件:

npm install vue-cli -g （vue-cli已更新到3.0时代，安装有变化，具体请百度查看器api）

npm install webpack -g

--构建打包成功之后，会在根目录生成 dist 文件夹，里面就是构建打包好的文件，通常是 \*\*\*.js 、\*\*\*.css、index.html 等静态文件。

如果需要自定义构建，比如指定 dist 目录等，则需要通过 config 进行配置。

1. 项目目录详解

|  |  |  |  |
| --- | --- | --- | --- |
| **标题** | **步骤/工具/目录/条目** | **操作/描述/详细** | **备注/参考** |
| 项目依赖工具 | webpack | webpack 是一个 JavaScript 应用程序的静态模块打包器(static module bundler) | [https://webpack.docschina.org](https://webpack.docschina.org/) |
| nodejs | Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。 | <http://nodejs.cn/> |
| element-ui | Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库 | <http://element-cn.eleme.io/#/zh-CN> |
| vue.js | Vue 是一套用于构建用户界面的渐进式框架 | <https://cn.vuejs.org/v2/guide/index.html> |
| es6 | ECMAScript 6.0（以下简称 ES6）是 JavaScript 语言的下一代标准 | <http://es6.ruanyifeng.com/#docs/intro> |
| lodash | lodash是一个 javascript 类库方便的提供对数据对象的操作 | <https://lodash.com/> |
| 项目安装 | 克隆项目 | git clone http://git.sfc.com/yuxang/warehouseService.git |  |
| 进入项目根目录 | cd ./warehouseService/Code/Frontend | 此目录为代码根目录 |
| 安装项目依赖 | npm install |  |
| 配置hosts | 127.0.0.1 local.sfcservice.com |  |
| 运行项目 | npm run dev | 开发环境运行初始化 |
| 项目目录详解 | /build | webpack 的配置目录 | 基本基于vue-cli创建项目生产的配置文件，不需要改动，包含了webpack的配置等 |
| /config | 项目的配置目录 | 开发及生产环境的配置目录，包括接口访问及本地hosts域名设置等 |
| /node\_modules | 项目所依赖的插件 | 所有依赖插件包 |
| /src | 项目代码的根目录 | 源代码 |
| /src/api | 所有接口请求配置目录 | 接口请求目录，按照业务分文件编码 |
| /src/assets | 静态资源目录 | 放置一些静态图片等 |
| /src/components | 项目公共组件目录 | 公共组件目录 |
| /src/directive | 项目指令目录 |  |
| /src/icons | 项目所有 svg icons | 系统采用的svg字体图标，使用参照：https://juejin.im/post/59bb864b5188257e7a427c09 |
| /src/lang | 国际化 language | 多语言目录 |
| /src/mock | 模拟数据目录 | 项目mock 模拟数据 |
| /src/router | 路由配置目录 | 开发环境及生产环境路由配置，左侧菜单的结构及图标、路由的跳转path及name全部在此目录下 |
| /src/store | Vuex 全局环境变量存储目录 | <https://vuex.vuejs.org/zh/guide/> |
| /src/styles | 样式目录 | 全局样式目录，基于scss开发 |
| /src/utils | 项目公共辅助js目录 | 全局公用方法 |
| /src/views | 视图目录 | 整个系统的页面全在这个目录下 |
| App.vue | 入口页面 |  |
| main.js | 入口文件 加载组件 初始化等 |  |
| permission.js | 权限管理 |  |
| static | 第三方不打包资源 | 不需要打包的资源插件可以放这里 |
| .babelrc | babel-loader 配置 |  |
| .eslintrc.js | eslint 配置项： ESLint 不管是多人合作还是个人项目，代码规范是很重要的。这样做不仅可以很大程度地避免基本语法错误，也保证了代码的可读性。  # 配置项 所有的配置文件都在 .eslintrc.js 中。 本项目基本规范是依托于 vue 官方的 eslint 规则 eslint-config-vue 做了少许的修改。大家可以按照自己的需求进行定制化配置。  比如：我个人或者项目组习惯于使用两个空格，但你可能觉得四个空格更顺眼，你可以做如下修改。 进入项目 .eslintrc.js 中，找到 indent，然后修改为 4 即可。 还有各种各样的配置信息，详情见 ESLint 文档。  在 v3.8.1版本之后，增加了eslint-plugin-vue来更好的校验 vue 相关代码。  默认情况下使用了最严格的plugin:vue/recommended来校验代码，若你觉得太严格可自行修改。  // https://github.com/PanJiaChen/vue-element-admin/blob/master/.eslintrc.js  module.exports = {  extends: ['plugin:vue/recommended', 'eslint:recommended']  //你可以修改为 extends: ['plugin:vue/essential', 'eslint:recommended'] } # 取消 ESLint 校验 如果你实在是不想使用 ESLint 校验，只要找到 config/index.js 文件。 将 useEslint: true 设置为 useEslint: false 即可 | |
| .gitignore | git 忽略项 |  |
| .travis.yml | 自动化CI配置 |  |
| favicon.ico | favicon图标 |  |
| index.html | html模板 |  |
| package.json | package.json |  |