# 目录说明

Build ：WEBPACK配置文件。打包时需要。含开发环境和生产环境。

公用配置文件：webpack.base.conf.js

开发配置文件：webpack.dev.conf.js

生产配置文件：webpack.prod.conf.js

Config ：NODEJS配置文件。有域名网址、端口等配置。如：API\_DOMAIN与API\_CONTEXT

Node\_modules：NodeJS开发依赖的所有包。npm install编译安装需要的包都在这里。

Server：服务器相关的一些文件，开发时没有用到。

SRC： 源码目录

Api . 所有的请求class都封装到了这个文件夹中

Assets 项目的资源文件存放文件夹

Components 所有的页面组件都在这个里面

Model 项目中一些用到的常量

Router 路由配置 （采用了history模式，base为h5为了兼容生产环境）

Scss 存放css相关的常用class与字体颜色

Store vuex文件相关都存放在这里( 用的不多，只缓存了头与底部的数据，避免重复请求)

Utils 开发中用的工具类（http-client 为所有请求的基类）

Src/App.vue 单独讲一下， rem的计算是写在这里了， 全局的回到顶部悬浮框也在这里

Src/Main.js 全局路由拦截，不过后端也做过拦截，详细可以看http-client

Static 也是存放静态资源的地方 （区别于asset的就是。这里放的是你引入的外部插件的静态资源，assets则是放自己项目的静态资源） ，该文件夹没有用到

Pakage.json 项目所有使用的依赖都可以在这里找到

# 需要的环境

Nodejs。 版本5及5以上。

# 开发工具

按个人喜好，推荐vscode。

# 开发测试步骤

1. 开发前修改：

a) 修改config/index.js  
 将dev下的host改成localhost,端口使用8090，本机运行测试  
b) 修改后台URL配置  
 将dev.env.js的API\_DOMAIN与API\_CONTEXT 改成后台路径  
c) 首次运行，使用 npm install 安装 node\_module

1. 运行

在项目根目录下运行CMD命令  
 npm run dev 该命令可以在 pakage.json已配置

1. 调试技巧
2. 浏览器断点调试:  
   在Webpack.dev.config 中将 devtool设置为source-map
3. 或者使用 Chrome的扩展工具 **vue-devtool** 来调试

# 项目正式发布步骤

1. 发布配置修改

a) 修改config/index.js  
 将build下的assetsPublicPath 改为‘./’，已改好。  
b) 修改后台URL配置  
 将prod.env.js的API\_DOMAIN与API\_CONTEXT 改成与dev.env.js一致。

1. 编译

在项目根目录下运行CMD命令  
npm run build

3. 编译完成后，查看是否成功，如成功，则会在根目录下生产 dist文件夹。将该文件夹下文件压缩打包后交由后台人员发布即可。