一 文件目录层级

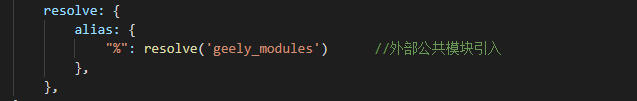


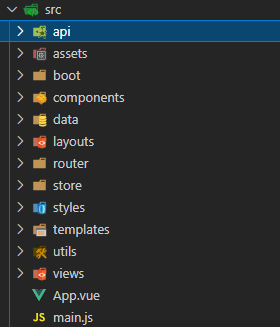
1.静态资源文件夹:public和src中的assets

<https://blog.csdn.net/Amnesiac666/article/details/111478209>

2.路径中的@和%

@表示当前src %表示geely\_modules 通过vue.config.js定义



3.src目录结构

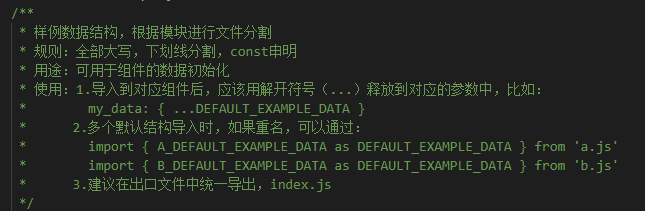
1.api:本项目中的api，主要和当前项目耦合度高的api定义，纯接口api定义在geely\_modules中的api

2.assets：静态资源文件，会通过webpack处理

3.boot:项目公共常用组件封装，包括geely\_modules中的公共模块导入。在main.js中调用。其中main中是最基础的组件，包括element,axios,auth(定义路由首位) ，plugins是其他组件

4.components:可复用组件，主要是当前项目中的不同页面可复用

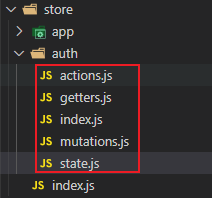
5.data:数据定义



6.layouts:定义项目框架组件(单独放在文件夹中，不放在components中)，router定义中一级路由的component只想layout.vue，实现每个页面都在框架下

7.router:vue-router封装，实际项目通过boot中的auth.js(定义路由守望)去重新更新routers:api获取用户路由，调用router文件夹utils中的函数

8.store: vuex实现(响应式、控制提交) 相当于一个网页上的简易数据库

a.state.js:定义存储的变量(state)

b.getters.js:获取state中的变量

c.actions.js:改变state中的变量，

d.mutations.js: 改变state中的变量,可同步，通过commit提交actions改变state中的变量

页面调用 通过 computed中定义

computed：{

auth() {

return this.$store.getters["auth/user"];

}

}

9.styles:个性化样式

10. templates:定义模板页面，主要是当前项目中 通过slot插槽

11.utils:通用工具函数定义，已转移至geely\_modules中，api中主要是与接口相关的函数

12.views:界面开发：不同模块分不同文件夹，每个模块中一般有index.vue，components文件夹，index.vue对外暴露，components中写一些index.vue中调用自用组件，比如一个典型的列表：

二简易开发流程

1. 定义路由，目前在mock中的router.json中手动定义，后期通过后台维护
2. 根据路由定义中的component在views建对应的文件，新建index.vue,components文件夹
3. 在当前index.vue中写页面，需要自定义的组件写在同级components中，可复用的组件在当前项目中的componengs或者geely\_modules中查找
4. 写的过程中需要用到的api在api文件夹中按当前文件夹名建立，用到的数据结构在data-defualt文件夹中按当前文件夹名建立

三mock假数据说明

目前api定义中使用mock假数据返回，参照geely\_modules api中的department