# Tip:基于vue-cli3的重构

1、安装vue-cli3(如果已安装则跳过)：

npm install -g @vue/cli

2、清除原项目下的文件。

注：需要保留的文件为，

1. src目录
2. index.html
3. static目录
4. package.json

3、创建vue-cli3项目：

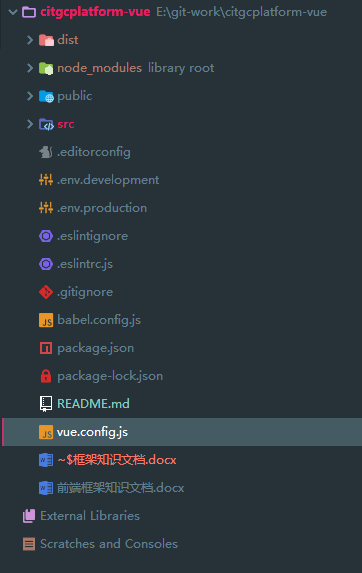
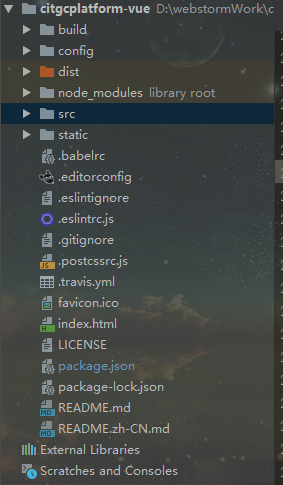
vue create my-project

4、将my-project中的除node\_modules、src、package.json

以外的文件拉到citgcplatform-vue中。

# 一、了解基础结构

目录结构（新版的改动）：



（旧版） （新版）

## 1、认识package.json项目配置文件

只介绍重要的字段意义，更多其他的package配置请查看：

<https://docs.npmjs.com/files/package.json>



name：项目名称。

version：当前项目版本。

scripts：npm运行命令。npm执行命令以’npm run’开始。 控制台运行时可直接输入npm run dev则以开发环境运行。npm run build:prod则进行生产环境的代码打包，打包的代码可直接发布。

lint-staged：eslint代码查检工具要检查的文件格式，当前只检查.js或.vue的文件，如果定义了则在代码提交到git仓库时会做检查，如不符合.eslintrc.js文件中定义的规则则提交报错，无法正常提交代码。

repository：指示本项目依赖的原官方项目git仓库地址，后续项目中的组件更新可通过该地址同步官方更新。

dependencies：项目的插件依赖包模块，通常是项目运行所必需的模块。

devDependencies：项目开发时的插件依赖包模块，只在开发时用得到，不会被打包到线上。

engines：描述node的版本需求。

安装项目依赖包：npm install， 如果添加了淘宝镜像也可cnpm install。该命令会下载dependencies和devDependencies中的相应包名所对应版本的包。



（新）scripts命令中只保留这三个，分别是运行开发环境、运行生产环境、运行代码

## 2、认识.eslintrc.js



该文件用于配置开发时的代码编写规范，最主要的是rules字段下的定义。rules中定义的规则会在我们编写代码的时候如不合规范会给出相应提示。

更多的具体配置规则请查看https://cn.eslint.org/docs/rules/

## index.html

Vue单页应用到的主入口文件，整个应用都在该html中。可在这里引入一些静态第三方资源。

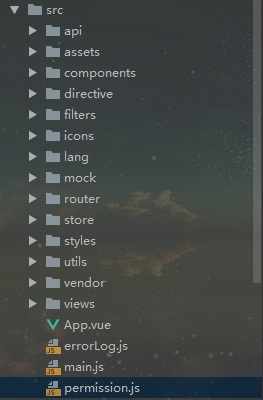
（新）index.html文件被移入到public目录下

## Build/config/node\_modules/static

Build下是webpack的配置文件；config下是应用的开发配置；node\_modules是项目安装的所有依赖包；static中多存放不需要通过webpack打包的静态资源。

（新）去除build目录，现在的项目配置直接在项目下的vue.config.js文件中做配置修改。

## src开发目录



api：存入应用中的接口定义。

assets：静态资源。

components：共用组件。

icons：图标库下载的文件，项目使用iconfont图标库。官网地址：https://www.iconfont.cn/

router：路由的配置。

store：vuex的状态管理配置。

styles：样式的文件目录。项目使用less预处理器编译css。

utils：项目中的常用工具类封装。

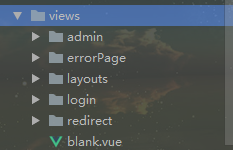
views：页面文件。

main.js：启动vue的入口文件。

permission.js：对每个路由的跳转作处理，通常验证是否有登录，是否能够进入相应路由页面。

# Src目录开发注意项

## views页面组件



按页面型模块划分页面目录。

layouts下存放每个系统的主布局入口。

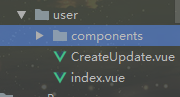
login登录页。

admin现有主系统页面。

errorPage处理系统中发生错误时的提示性页面。

blank.vue多级路由的中转页面组件。

每个模块下如果有对应的公用组件，则在该模块下新建components文件夹，存放该模块下的公用组件。



## 路由

本项目的路由配置结构不同于官方的配置，是基于项目需求而改造过的。

正常配置：





constantRouterMap中放的是整个路由中常规的不需要动态获取的路由，通常是登录页、错误页等不需要权限的页面。

改造的路由：

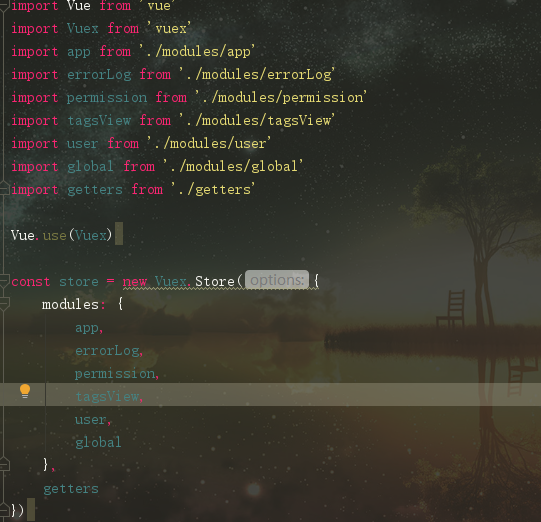
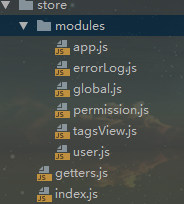
Blank是一个空白组件，作用是用于当多级菜单时，通过它作为容器显示。

menuIdComponents里的路由都是异步路由，用于通过和接口获取的菜单数据作比较得出最终应该展示的路由页面。该对象下的每个字段key对应于接口菜单中的key值，只有在接口菜单中发现了相应key值的路由才会被加载到页面中。每个路由的path路径如果以‘/’开头则默认作为一级路由展示，否则会作为子菜单。

关于key：例如在遍历菜单接口时发现接口中的key字段与我们路由menuIdComponents中的某个key相符，则认为该路由需要被加载。

为什么不按官方的配置来？因为项目中的菜单是可以随意放置在任一菜单下的，位置是不固定的，所以需要把打散的路由拼成菜单的结构。

## Store状态管理

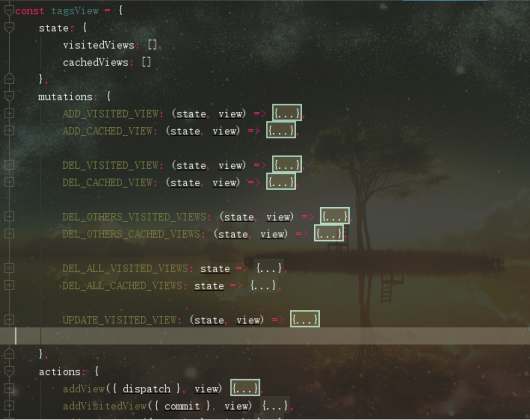


这里存放的数据属于全局的，或者是跨组件层级太多的时候才需要放到这边。

通过store/index.js导出整个store。

getters.js用于从store/modules中读取每个块的state，其他作用可以认为像vue的计算属性一样，可以把计算值缓存在应用中，只有当它的依赖值发生了改变才会被重新计算。

每个store modules 的写法定义：



state：当前store modules的数据。

mutations：提交保存state的操作，所有的更改state操作都从这里执行，是同步操作。

每个方法名大写，单词间以‘\_’隔开，第一个参数为当前modules下的state，后面的参数为调用地方传的参数。

actions：多用于异步操作，如调用接口。mutations当中只能以同步进行，所以actions专用于异步操作。

每个方法名骆峰写法，第一个参数为object，提供{commit,dispatch,state}常用参数，后面的参数为调用地方的传参。

dispatch(‘actions’,params)发起其他的actions，参数一为actions中对应的方法名，参数二为提交的值；commit(‘MUTATIONS\_NAME’,params)用于提交至mutations，参数一为mutations中对应的方法名，参数二为提交的值。state为当前modules的数据

在vue中

发起actions：this.$store.dispatch(‘actionsName’,params)

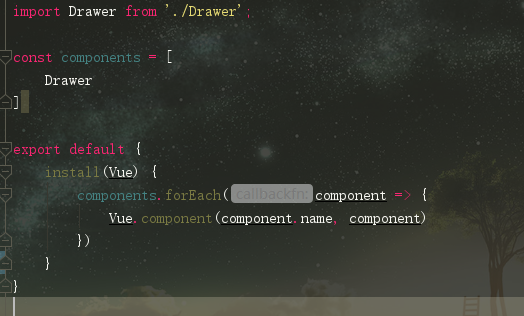
提交mutations：this.$store.commit(‘MUTATIONS\_NAME,params)

获取state：this.$store.state

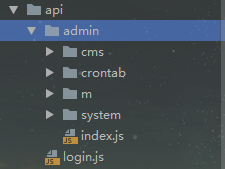
获取getters：this.$store.getters

## components公用组件

只存放全局公用的组件。如果要注册不需要导入的全局组件，则在components/index.js中载入对应的组件：

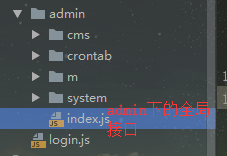
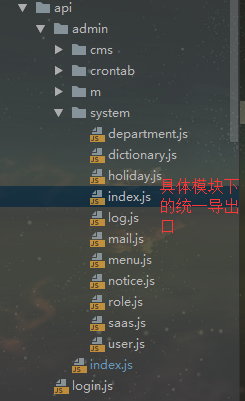


## api接口定义

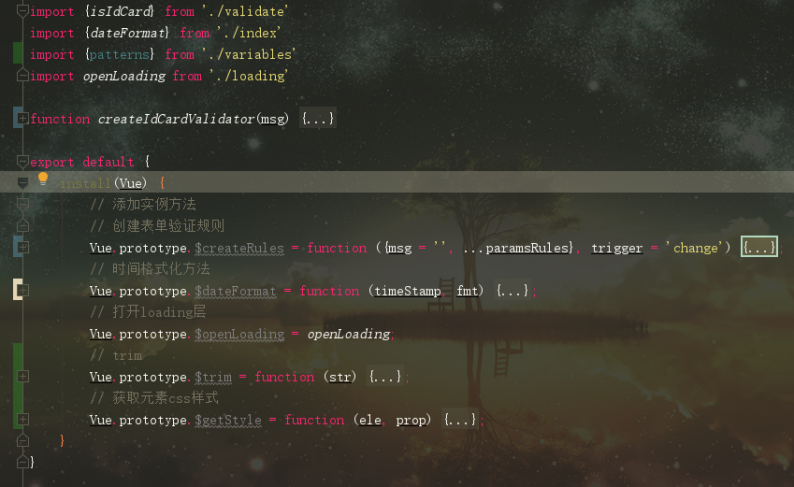
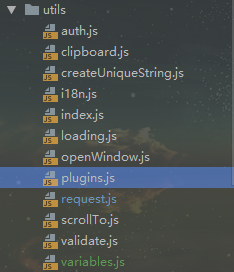


定义与后台交互网络请求地址，按页面功能模块区分，分别对应views文件夹下面的页面。

每块接口由模块下的index.js统一导出。每个大模块下的index.js属于该模块下的全局api。



## utils工具



此目录下放的是全局常用的方法，

在plugins.js中注册vue实例的全局插件，每个插件块都应该有一个install方法，install方法中则可以将要注册的方法绑定到Vue.prototype上，方法名以$开头。

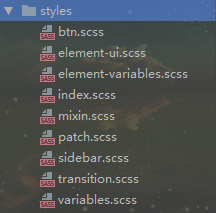
request.js中定义接口请求的axios实例，全局中调用的接口工具。

variables.js定义全局变量。

loading.js打开loading层工具。

index.js主要方法工具导出口。

## 7、styles样式



element-ui.scss对修改element-ui样式的修改。

element-variables.scss为覆盖element-ui变量，如果要更改element-ui默认变量则在这里修改。

index.scss是我们自己的样式。

patch.scss对第三方样式的修改。

variables.scss样式变量。

## 8、Icons图标

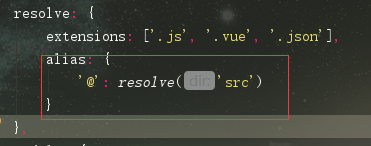
从图标库下载的图标所在地。

图标库地址：<https://www.iconfont.cn>

图标使用类名：iconfont icon-xxx

# 开发代码注意

1、项目有引入模块时，路径中可以采用别名的方式，在webpack.base.conf.js中可以配置引用的别名名称，避免文件层级嵌套太深时路径查找麻烦。要配置其他别名在这里配置。



（新）别名配置在vue.config.js中修改

1. 所有的组件都是vue单文件组件形式存在，每个组件中都要有name字段，用于给当前组件添加标识(name并不是vue中必须的，我们建议要有name)。
2. 表格的公用组件定义，组件传入的属性：fetchDataFn:接口请求的地址，是一个axios请求；query:请求接口的查询参数；columns:列信息。 columns中key:对应数据中的字段名的值；label:列名；align:对齐方式；width:列宽；render:自定义显示内容。 render中要返回的是组件或纯文字的字符串，第一个参数为当前行的数据，第二个参数为vue提供的jsx创建组件的方法。render返回的组件都采用jsx创建方法，具体使用方法可参考vue官网提供的说明。

