怎么运行?

1. svn下载该目录代码
2. Cd进入该目录
3. 运行命令 npm i (需要安装nodejs: https://nodejs.org/en/download/)
4. 1 运行命令 npm run dev进入开发模式

4.2 运行命令 npm run build生成发布文件,在dist目录

相关链接:

Vue:https://cn.vuejs.org/v2/guide/

VueRouter(路由):https://router.vuejs.org/zh-cn/

Vuex(状态管理):https://vuex.vuejs.org/zh-cn/

vee-validate(表单校验):http://vee-validate.logaretm.com/

mint-ui(第三方UI):http://mint-ui.github.io/docs/#/zh-cn

ES6入门(新标准的js语法):http://es6.ruanyifeng.com/

其他:

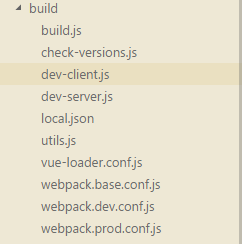
muse-ui:http://www.muse-ui.org/#/install

element-ui:http://element.eleme.io/#/zh-CN/component/installation

vue-devtools.crx是chrome的vue调试插件

vuex.create.xlsx是用来生成vuex的固定代码

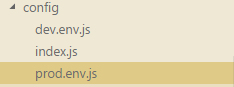
-------------文件目录介绍



Webpack的相关文件

dev-server.js:自带的反向代理,即前端的请求可以通过这个来转发请求,解决跨域问题,这个只在开发状态下有效

Local.json:本地数据源



一些配置,我也没怎么用过,index.js有些参数可能会用到;

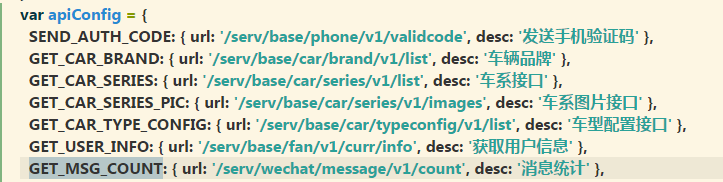
Demo目录没用,可以删

Dist 是发布文件的生成位置

Node\_modules:npm依赖文件的下载路径,刚开始运行npm i下载的东西都保存到这里,

Src:我们的开发文件都放在这里

Api>api.js:访问后台的方法都放这里,接口的url都配置在



Assets:存放静态资源比如图片

Components:组件的存放目录

Pages:页面的存放目录,其实pages和Components都是vue文件,没什么区别

Router:路由配置的文件,新建一个路由地址的方法

1.导入组件/页面 **import index from '../pages/index';**

意思是从pages/index.vue导入为变量index

2.建立路由配置:**{ path: '/index', meta: { title: '首页' }, component: index },path是路径,即浏览器访问的路径,title是页面header中显示的标题,在main.js的router.beforeEach方法中变换标题**

Store:vuex文件的目录

Index.js:vuex的入口文件,初始化了state,和新建vuex实例,关于state可以简单的理解为全局变量,但这个全局变量只能在mutation.js中修改,不然会报错;

mutation-types.js:定义了state,action,api,mutation之间关系的关键字

**export const GET\_CLUE\_RECORD='GET\_CLUE\_RECORD' ;**

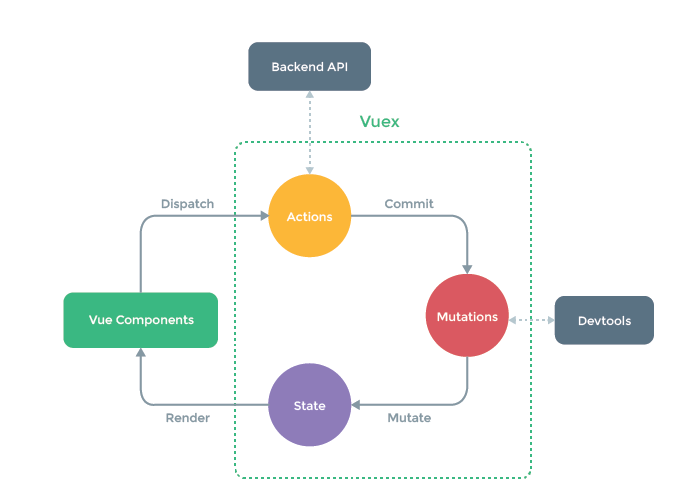
比如这个,那以上4个东西都叫**GET\_CLUE\_RECORD(state有个这名字的变量,action/api/mutation中有这个方法)**

**Action.js:在vue文件中通过调用action来发起一个请求**

**Mutation.js:在这个文件中对state进行修改**

**Getters.js:定义一些属性,这些属性是一个带有返回值的方法,当依赖的state变化,会重新执行这个方法;**

**以下是他们之间的关系:**



简单的来说,在页面中通过dispatch发起一个action,action会调用api请求数据,请求成功后,将获取到的数据给mutation,mutation对state进行更改,state更改后,依赖于该state的vue会重新渲染;

除非一些变量要全局访问,比如该项目中的用户信息,不然不建议使用store,因为比较复杂;一般的获取数据可以直接通过调用api.js来请求数据:

**import api from '../api/api.js';//引入api**

***//单个请求***

**this.getData([**

**{ type: types.GET\_CAR\_SERIES\_PIC, param: { dlrCode } }**

**])**

**.then(res => {**

***//请求成功获取到数据***

**console.log(res)**

**})**

**.catch(err => {**

***//获取数据失败***

**});**

***//多个请求,一次性返回数据,跟单个请求的区别只是getData的参数数组长度是多个***

**this.getData([**

**{ type: types.GET\_CAR\_SERIES\_PIC, param: { dlrCode } },**

**{ type: types.GET\_BRAND, param: { dlrCode } },**

**])**

**.then(res => {**

***//请求成功获取到数据***

**console.log(res)**

**})**

**.catch(err => {**

***//获取数据失败***

**});**

**Style:存放样式的地方,可以放styl文件**

**Utils:一些公共方法,目前只有util.js和vueValidate.js有用**

**App.vue:项目的顶级结点,所有vue文件都挂载到<router-view></router-view>**

**这是路由的出口,即渲染结点;**

**一些公共的请求可以放在这,比如请求用户信息;**

**Main.js:项目的入口**

**Index.html:项目的入口html**