目录

[1引入轮播图 1](#_Toc23615096)

[2修改底部导航栏的图标 2](#_Toc23615097)

[3解决transition的项目问题 3](#_Toc23615098)

[4九宫格 4](#_Toc23615099)

[5新闻列表 4](#_Toc23615100)

[6定义全局过滤器 6](#_Toc23615101)

[7改造router-link to为:to和路由传值 6](#_Toc23615102)

[8.公共局部组件引用和父子组件传值 7](#_Toc23615103)

[9.get和post 8](#_Toc23615104)

[10.解决导入mui.js的babel严格模式 9](#_Toc23615105)

[11.解决mui头部滑动组件不能滑动 9](#_Toc23615106)

[12.不能阻止滑动默认内部事件 10](#_Toc23615107)

[13.设置懒加载样式 11](#_Toc23615108)

[13.引入mui的头部scroll组件导致底部无法切换路由组件 11](#_Toc23615109)

[14.安装vue-preview 12](#_Toc23615110)

[15.使用编程式导航 14](#_Toc23615111)

[16.抽离swipe做成公共局部组件 14](#_Toc23615112)

[17.加入购物车的小球 15](#_Toc23615113)

[18.根据库存控制mui的numbox组件最大值 16](#_Toc23615114)

[19.vuex的使用 17](#_Toc23615115)

[20.判断路径显示返回 20](#_Toc23615116)

[21.全局插件组件,全局组件,局部组件 20](#_Toc23615117)

[22. vue-router使用 21](#_Toc23615118)

[23.基于vue第三方必须use注册和new实例到vm挂载 22](#_Toc23615119)

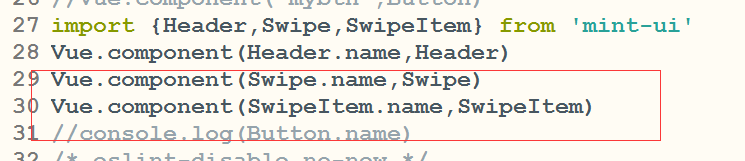
[24.每个vue文件的导包是独立的 23](#_Toc23615120)

[25.工程目录结构和打开必要的文件 24](#_Toc23615121)

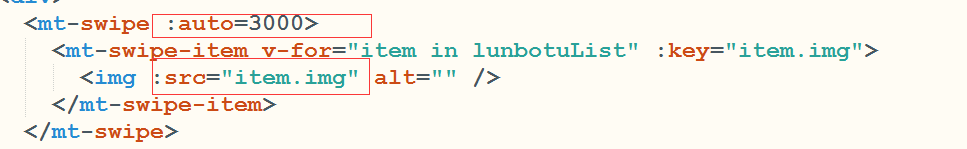
onlineLive-vue-开发细节

## 1引入轮播图

从mint-ui导入Swipe和SwipeItem

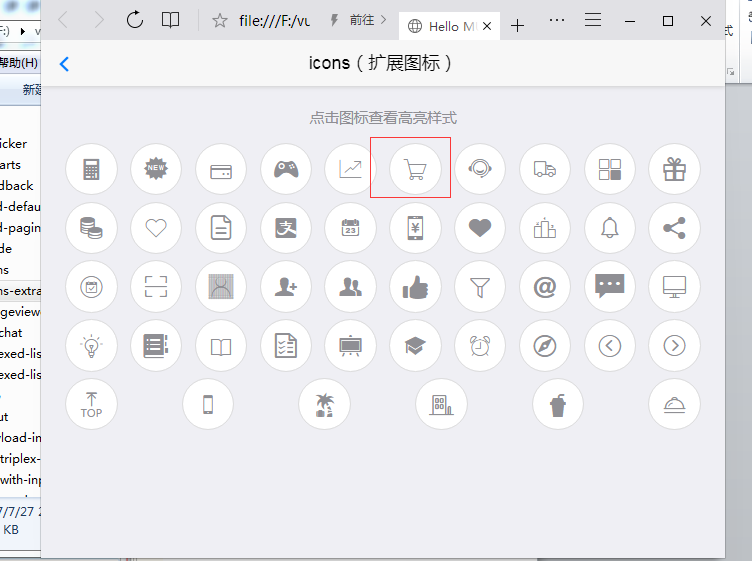


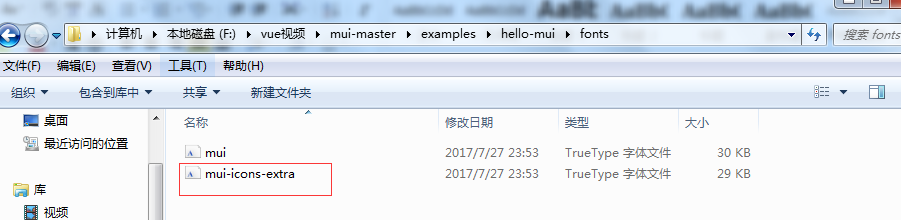
绑定时间 :auto=3000 绑定img :src

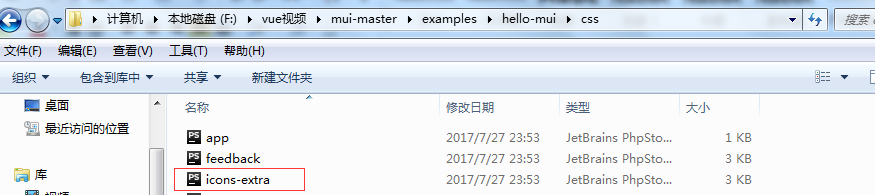


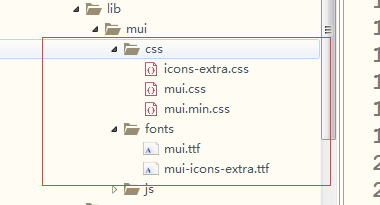
## 2修改底部导航栏的图标

需要引入字体图标和样式，购物车图标









## 3解决transition的项目问题

内容从下到上进入 在激活类

position：absolute//由于动画过渡，会将内容缩小再变大

带滚动条

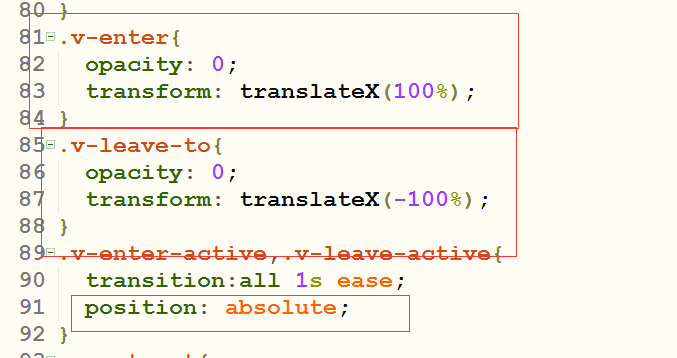
overflow-x:hidden;//因为前面一个内容还在，而下一个还在定位在左侧，所以有滚动条

设置入场和离场效果一致，会给用户理解不了页面和空白期

.v-enter{opacity:0;transfrom:translateX(100%)}

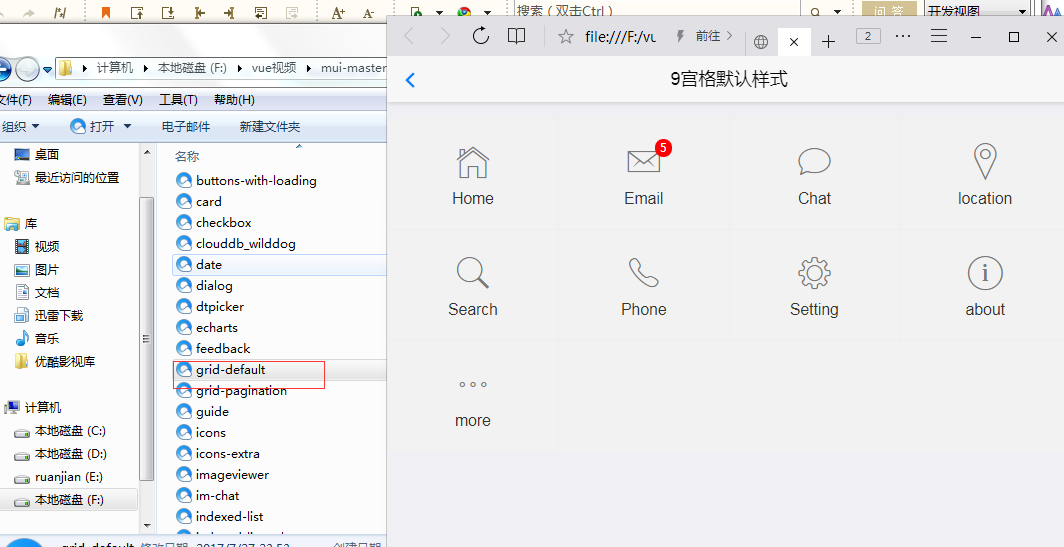
.v-leave-to{opacity:0;transfrom:translateX(-100%)}

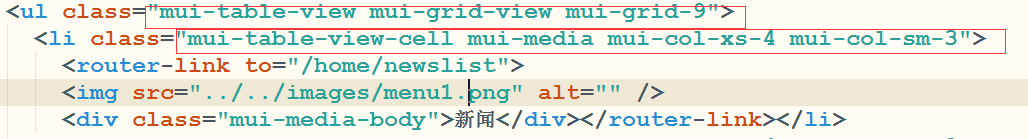
//入场和离场相反方向, 取盒子百分百



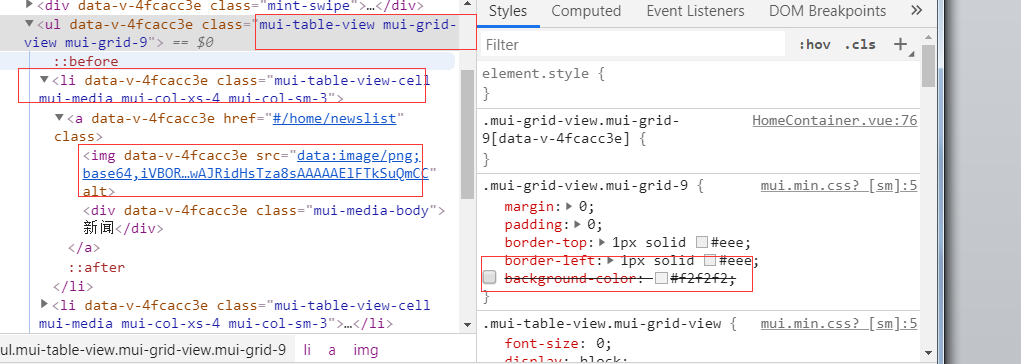
## 4九宫格

找到对应的文件，查看源码，把相关的代码拷贝



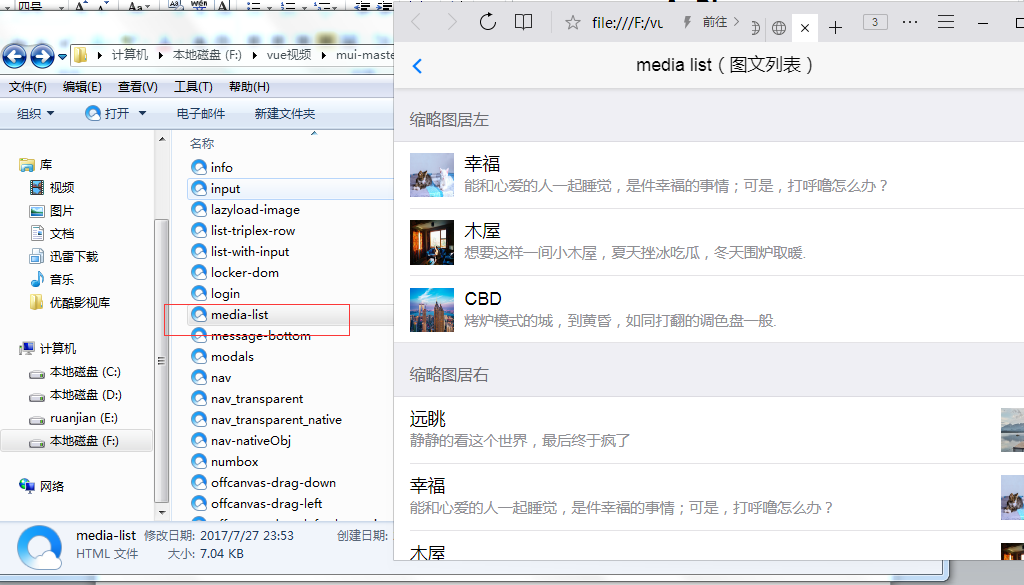


修改原本的边框线、颜色和图标大小，直接控制台查看

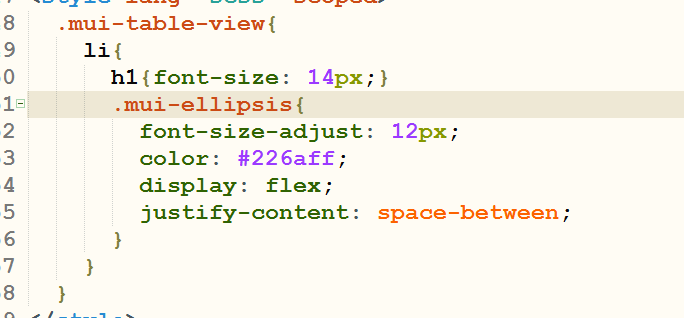


## 5新闻列表

找到对应的文件，查看源码，把相关的代码拷贝



List底部修改布局



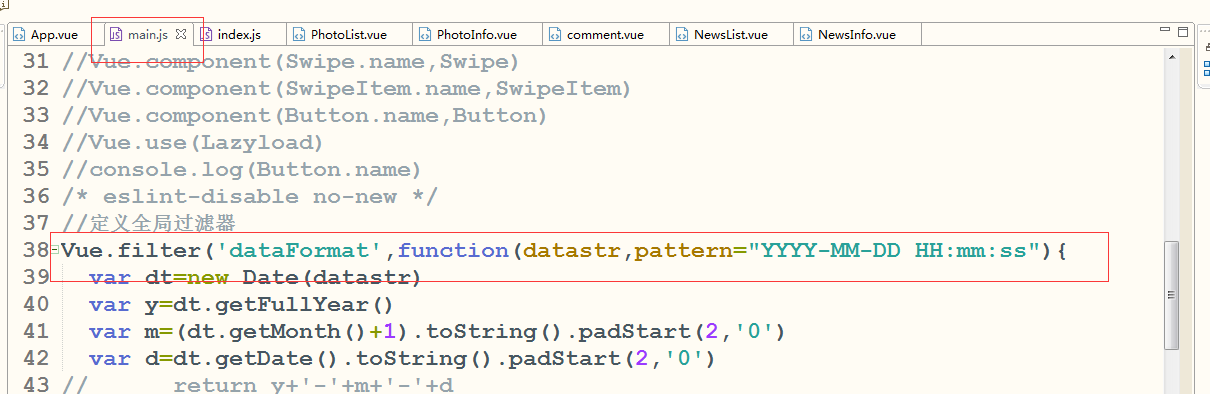


中文：的bug

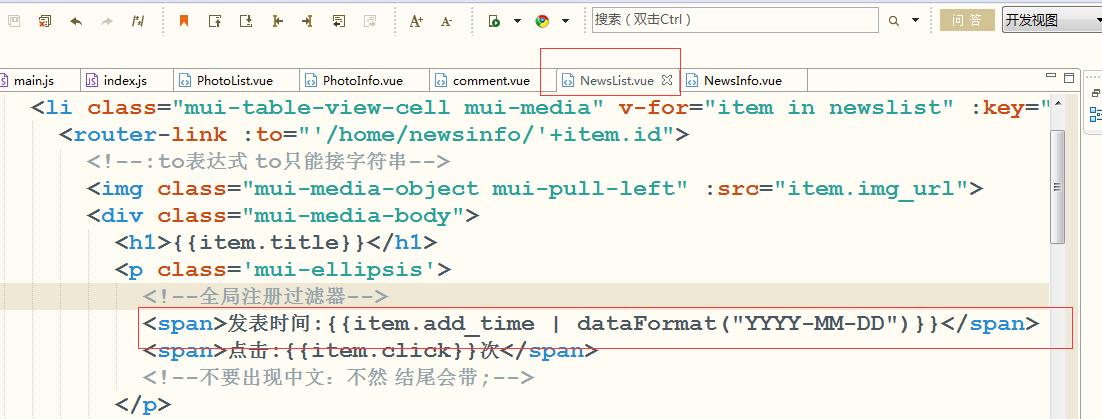
如果使用中文冒号，结尾会出现;

## 6定义全局过滤器

Vue.filter(‘filtername’,function(arg1,arg2=’value’){})//如果没有赋值，arg2的value为默认值, arg1为管道符前面的值



可以在路由组件引用{{ time | filtername(‘YYYY-MM-DD’)}}



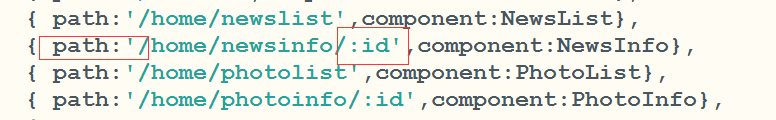
## 7改造router-link to为:to和路由传值

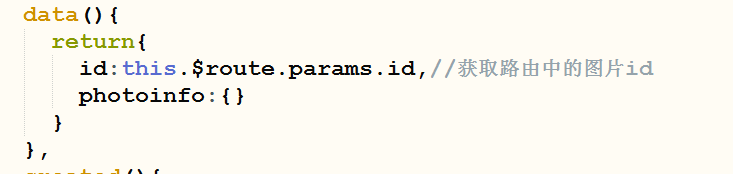
:to表示可以使用表达式，字符串用’’包起来,+进行连接

路由规则path可以指定参数:id ，然后:to 在对应的/位置赋值

to只使用字符串

获取路由参数的id，$router.params.id







## 8.公共局部组件引用和父子组件传值

在父组件页面导入子组件 import name from ‘’

注册子组件 components:’comment-box’:name//记得是加s

Template的组件里,属性绑定:propsid=’dataid’

子组件引入 props:[‘propsid’],可以直接this.propsid引用





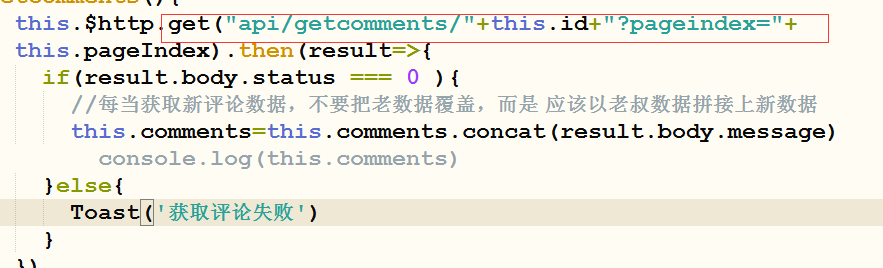




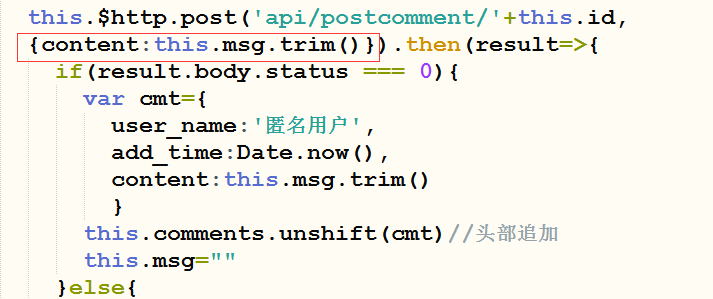
## 9.get和post

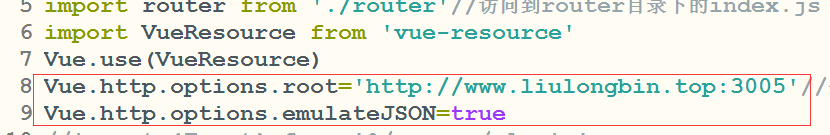
全局配置域名和端口

get显示传值 适合路由参数传值和查询传值



post隐藏传值、多种传值( 文件、表单) 全局配置post数据类型





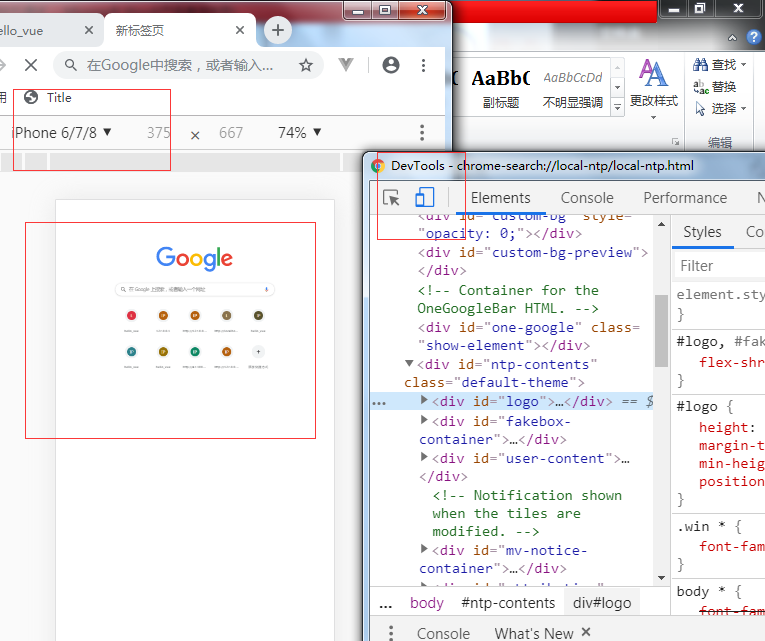
## 10.解决导入mui.js的babel严格模式

参考webpack文档

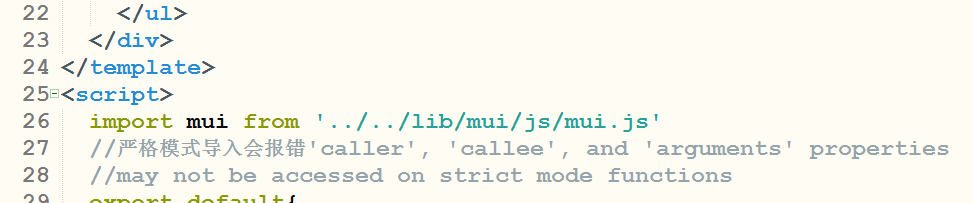
## 11.解决mui头部滑动组件不能滑动

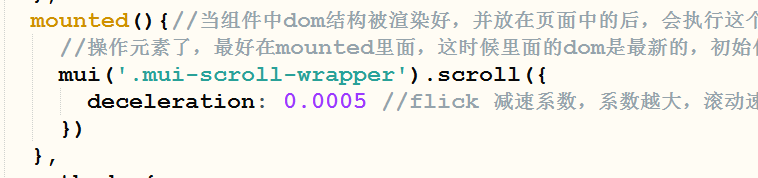
调试时，不要从pc切换到手机，这样mui.min.js文件等于没加载

打开谷歌浏览器切换到手机模式，再访问首页

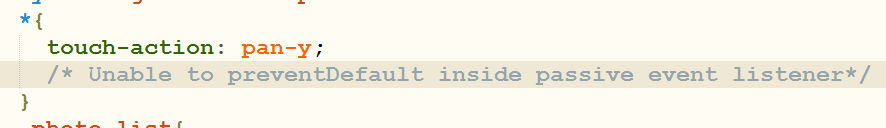
导入mui.js文件，初始化mui应在mounted()//因为此时组件渲染好了，这里等于局部导入了





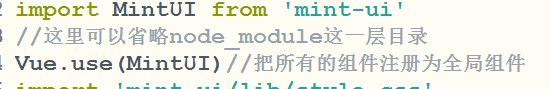
## 12.不能阻止滑动默认内部事件

只允许pan-y



## 13.设置懒加载样式

由于要使用到mui的懒加载Lazyload,只能全局引入,不然不生效

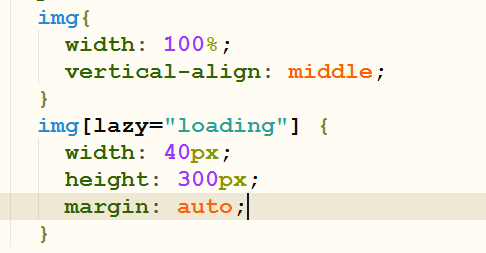




Img标签定义属性v-lazy

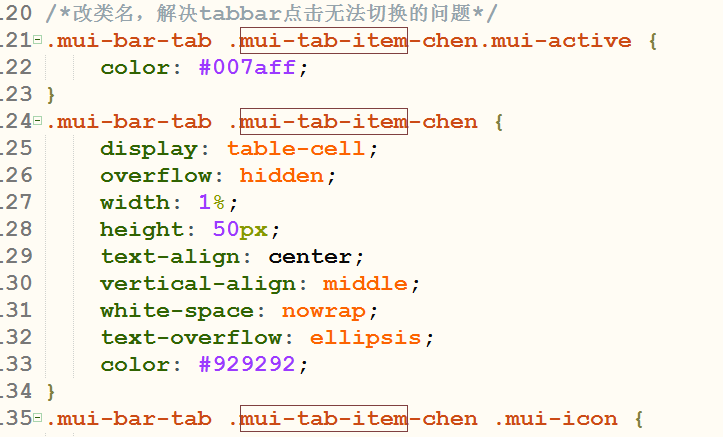


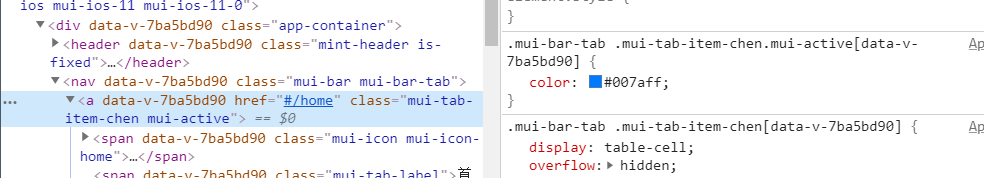
未加载前的样式

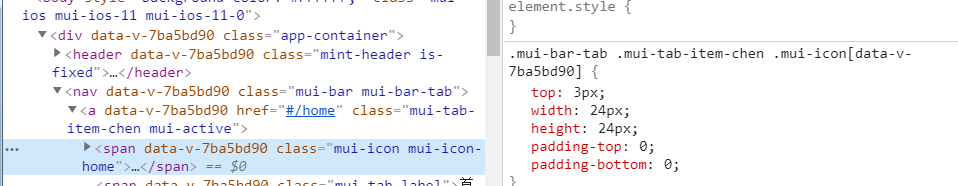


## 13.引入mui的头部scroll组件导致底部无法切换路由组件

复制有用到mui-tab-item类的样式，mui-tab-item-name扩展



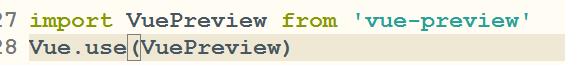




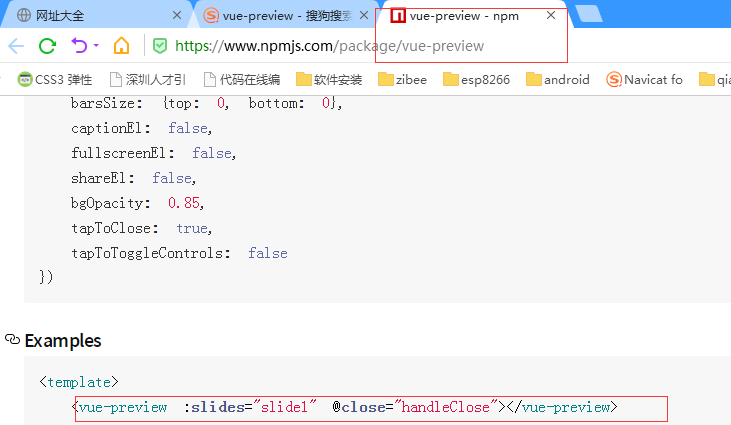
## 14.安装vue-preview

npm install vue-preview –S

全局引入

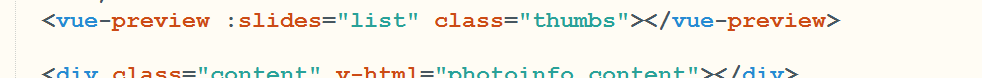


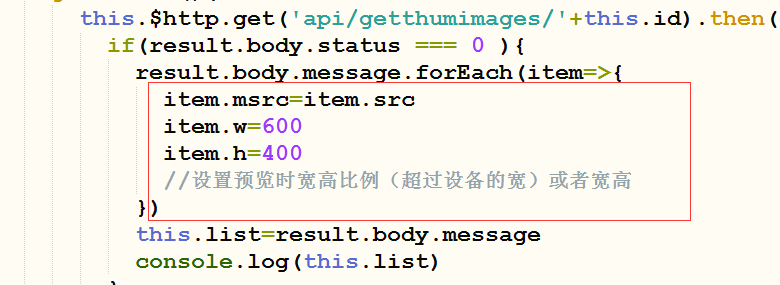
Github或者npm找到vue-preview



模板导入vue-preview

由于没有需要缩略图而案例没有提供，定义一个类进行改造，同时msrc要赋值src







## 15.使用编程式导航

区别分好$route和$router

$route存放路由参数的对象

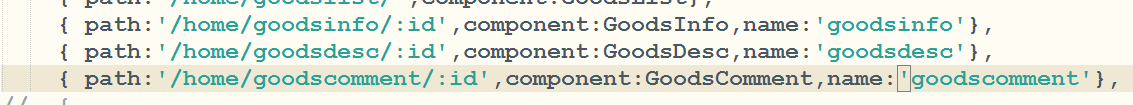
$router路由导航管理对象，可以执行一些追加、跳转，跟location差不多

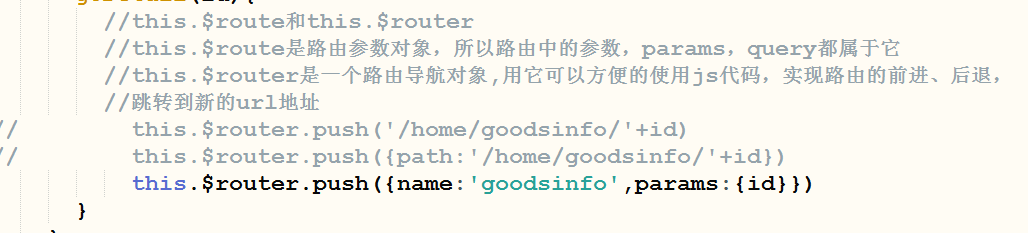
可直接使用h5标签和第三方组件跳转，在js执行

常用形式 name对应路由配置的routername’

this.$router.push({name:’ name’,params:{id}})

{path:’’,component:’importname’,name:’routername’}

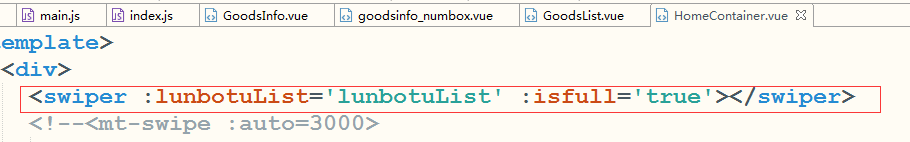


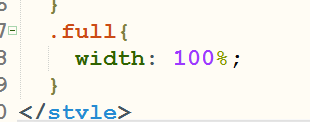
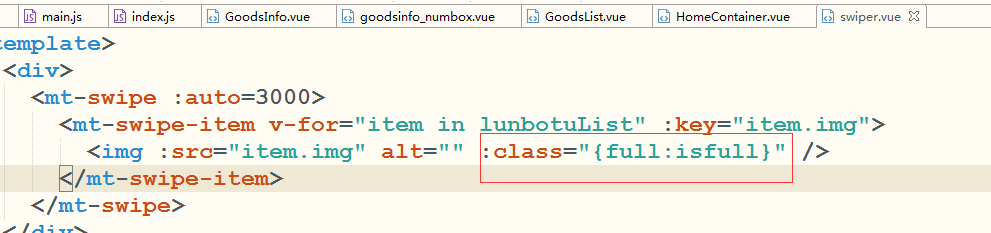


## 16.抽离swipe做成公共局部组件

由于首页的轮播图跟商品详情的图片展示有区别，首页是全部占满，而商品详情页的图片是高度占满，宽度自适应

在父组件绑定一个控制宽度的类bool值





## 17.加入购物车的小球

解决分辨率不同、滚动位置

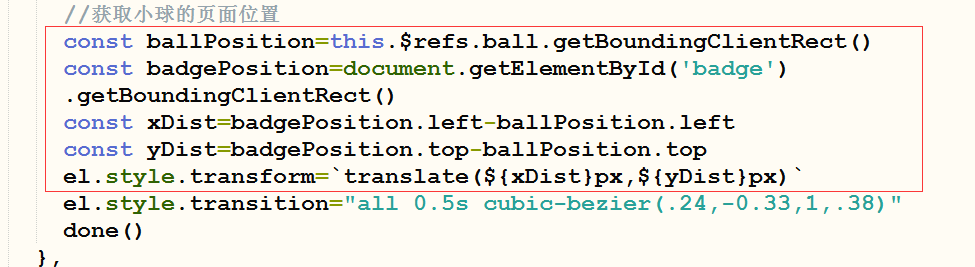
拿取徽标的左头位置减去小球的左头位置

小球是在详情页，可以通过$refs拿到

而徽标只能通过操作dom

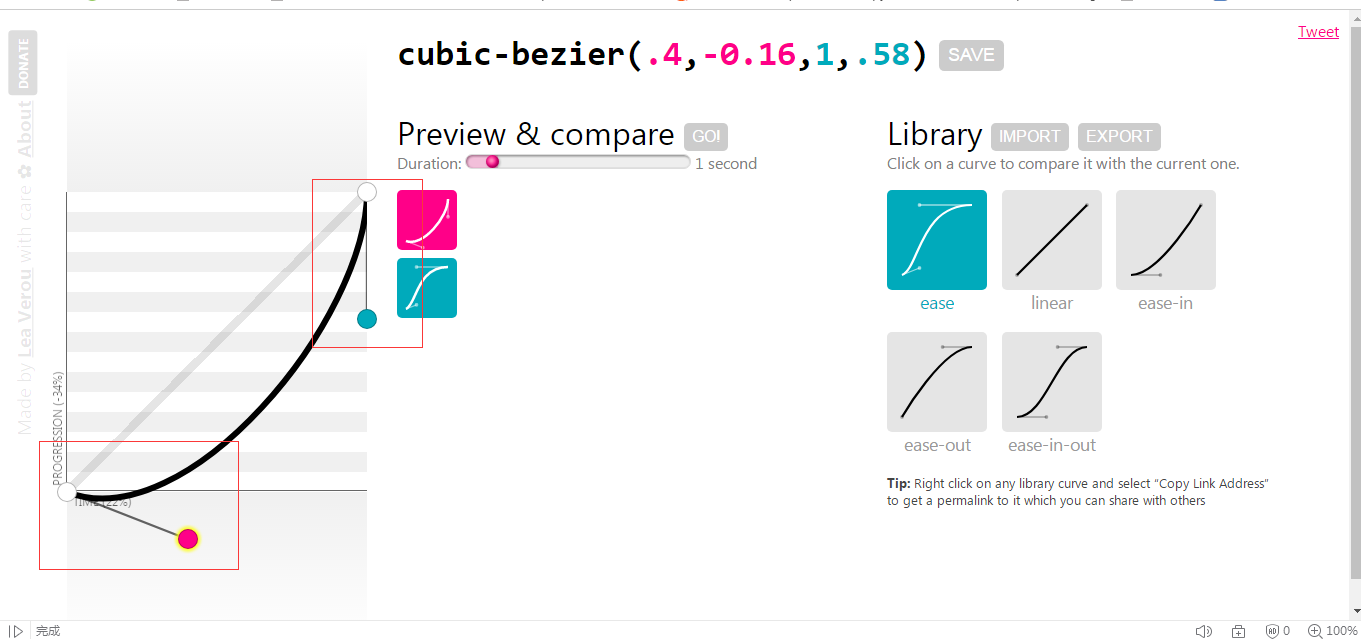
getBoundingClientRect()获取左头位置

使用模板字符串进行拼接



贝赛尔曲线cubic-bezier

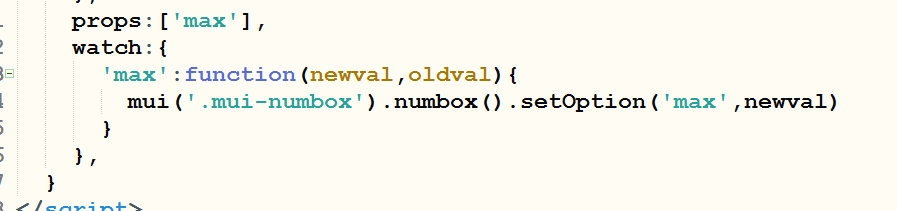
红色为起始

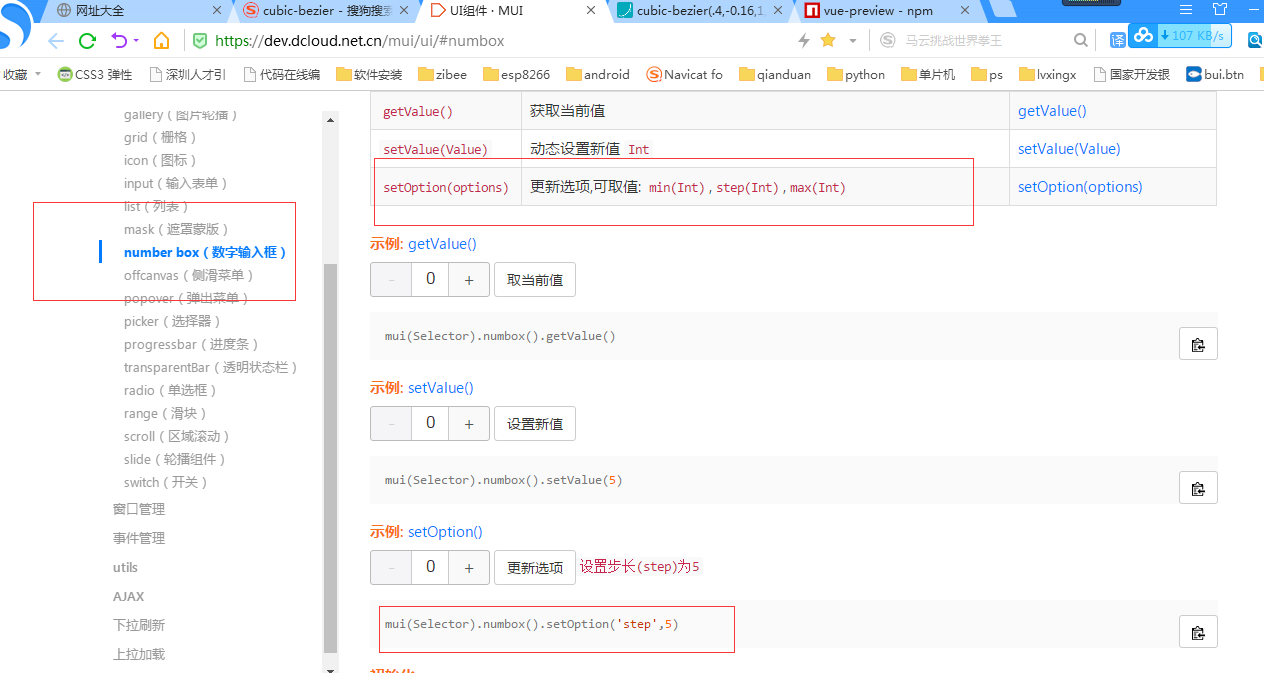


## 18.根据库存控制mui的numbox组件最大值

Mui的numbox的限定最大值通过获取父组件异步请求的库存量，所以没法在mui渲染好numbox时用上

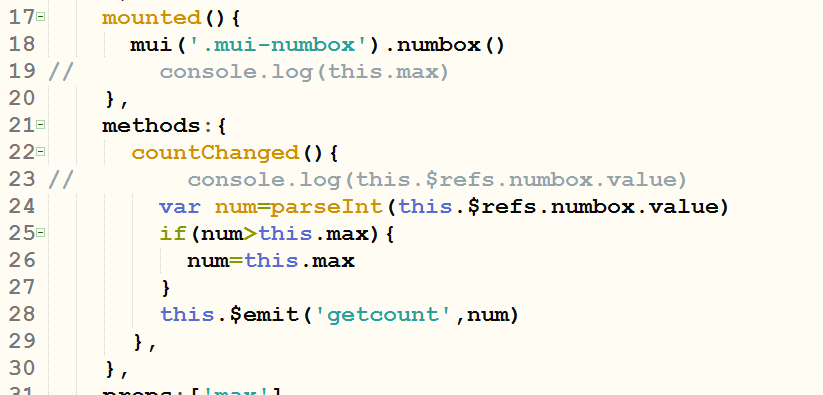
只能numbox组件里wacth监听，父组件绑定属性，同时要借助mui的api来设置



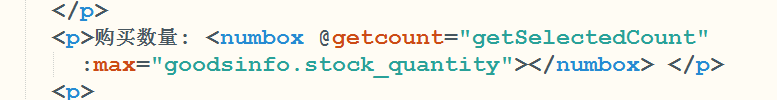


同时numbox组件改变值的变化，要时刻反馈到父组件，采用numbox组件调用父组件方法来传值，传值作判断

Numbox组件



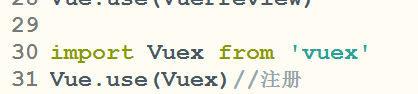
父组件



## 19.vuex的使用

npm install vuex –S

全局注入一个插件



Vm实例挂载仓库store

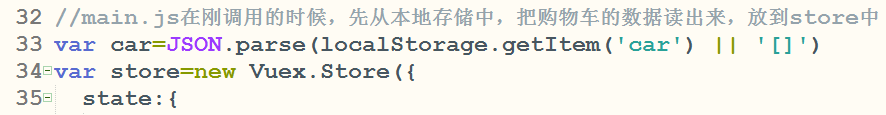
var store=new Vuex.Store//这样可以通过vm的this访问到



里面有三个对象

state//存储所有的仓库数量,里面的数据可以从本地存储获取json

数据JSON.parse(loaclStorage.getItem(‘’name) || ‘[]’）

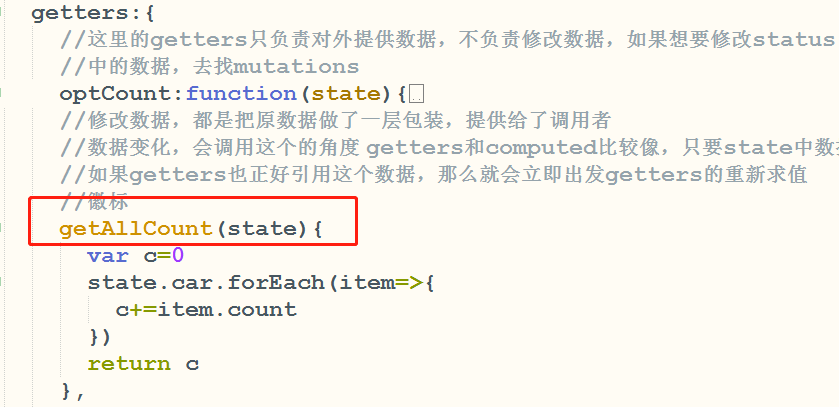


mutations//定义所有操作数据的方法,加入修改删除数据都要更新本地存储，只可以传两个参数，但是第一个参数是state，要想传两个参数以上先包装成对象

localStorage.setItem(‘name’,JSON.stringify(state.car))

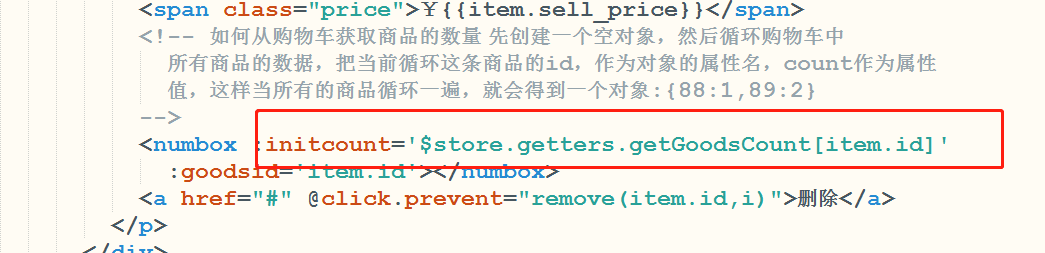


getters//定义所有获取数据的方法，作用是computed监听数据变化并实时更新，双向数据绑定，可以像filter 在内部return数据让调用位置使用, 不可以传参数



调用state、getters，$store.state/getters



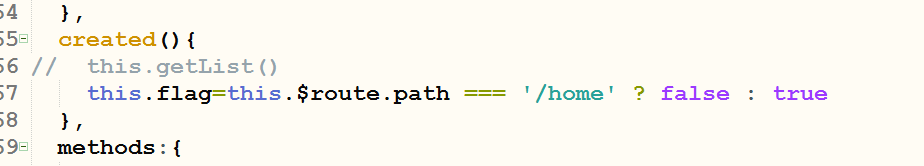


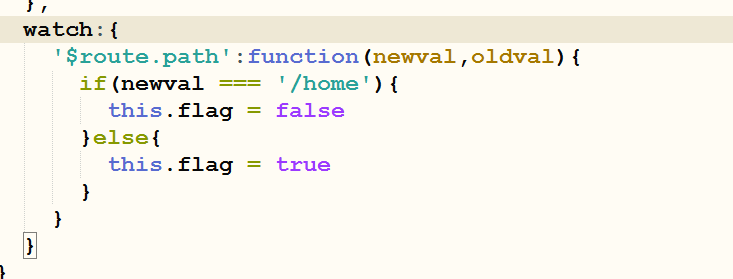
调用mutations ,$store.commit(‘name’,data)



## 20.判断路径显示返回

页面刷新和url变化监听wacth $route.path





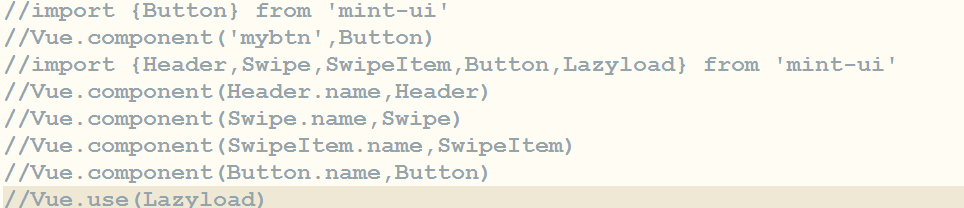
## 21.全局插件组件,全局组件,局部组件

Import ‘模块名/组件名’ from ‘模块名’

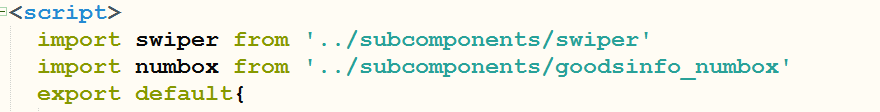
全局插件Vue.use(‘模块名’)

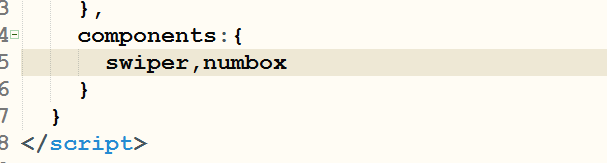


全局组件Vue.component(‘组件名’)



局部组件{components:{组件名‘’}}



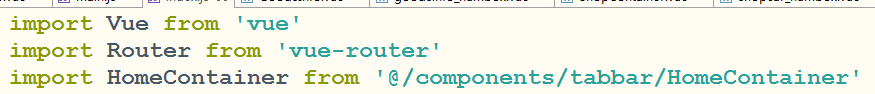


## 22. vue-router使用

创建一个router目录并创建index.js文件来存储路由配置

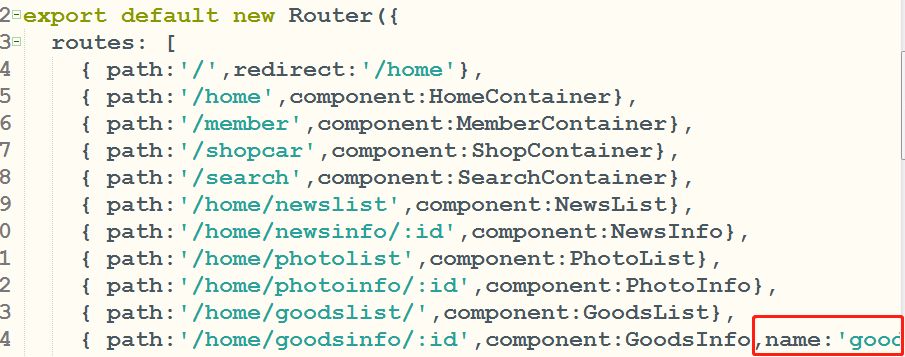
导入vue-router和vue

导入.vue模板文件 @表示src目录



注册vue.use

New Router实例配置routers的path、redirect、components、name(用于编程式导航)



## 23.基于vue第三方必须use注册和new实例到vm挂载

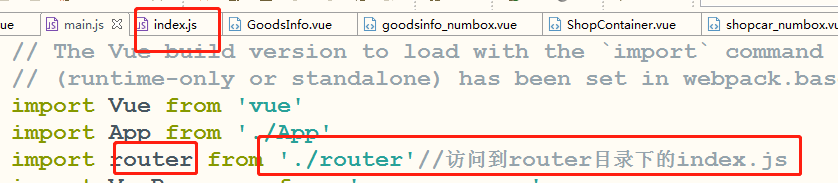
凡是涉及到new实例后的对象要想在vm的this访问到，必须在vm实例内挂载

1.注册Vue.use() 全局注册必须的(除了ui组件)

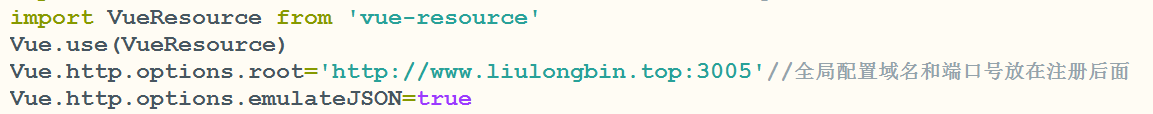
2.挂载new Vue{} 参考下面

目前有vue-router的配置对象router、vuex仓库对象store

2.1Router是index.js暴露的new对象



2.2Vue-resource由于配置是通过this.http，不需要new实例，则不需要挂载

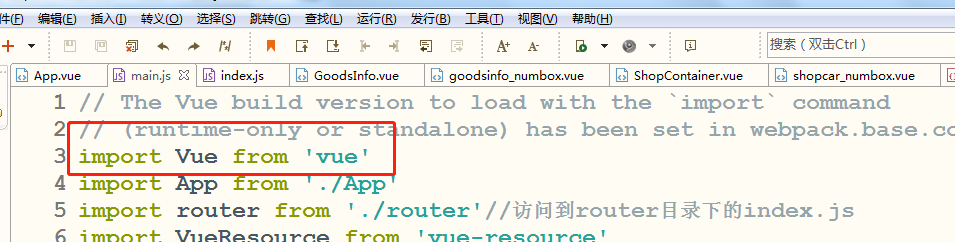


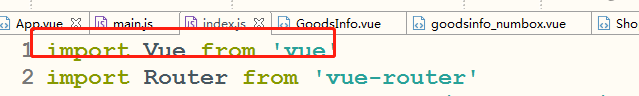
2.3由于mint-ui是组件，use会引入全部的组件，如果只需要引入某个组件通过vue.component全局注册组件同时配置自己想要的组件名



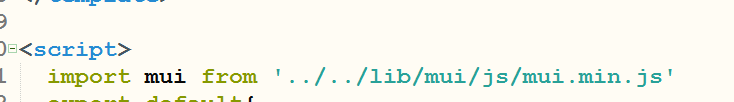
## 24.每个vue文件的导包是独立的

Main.js和路由的index.js vue都要各自导入





这里导入mui.min.js不会被其他文件使用



## 25.工程目录结构和打开必要的文件

