## 框架新手入门：

Vue官方文档（中文）：<http://cn.vuejs.org/guide/>

Webpack官方文档（英文） <https://webpack.github.io/docs/>

英文不及格的建议看 <https://github.com/vikingmute/webpack-for-fools/tree/master/entries>

讲的很不错，全部熟悉了也可以入门了。

Vue-router官方文档（中文） <http://router.vuejs.org/zh-cn/index.html>

Vue-resource（github） https://github.com/vuejs/vue-resource

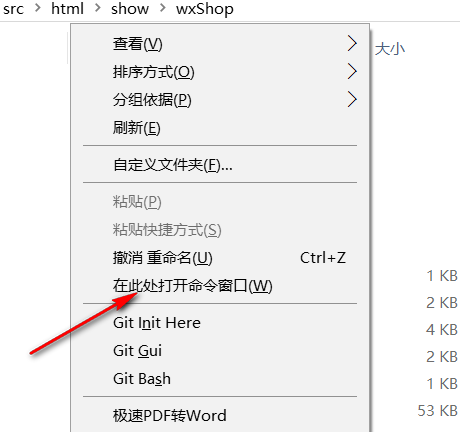
Vuex官方文档（中文） <http://vuex.vuejs.org/zh-cn/index.html>

Gulp官方文档（中文） http://www.gulpjs.com.cn/

博城云商项目使用技术：

1. vue
2. webpack
3. vue-router
4. vuex
5. vue-resource
6. gulp

运行

在当前项目文件根目录下按住键盘shift键不放，在文件夹里面的空地方点击鼠标右键，

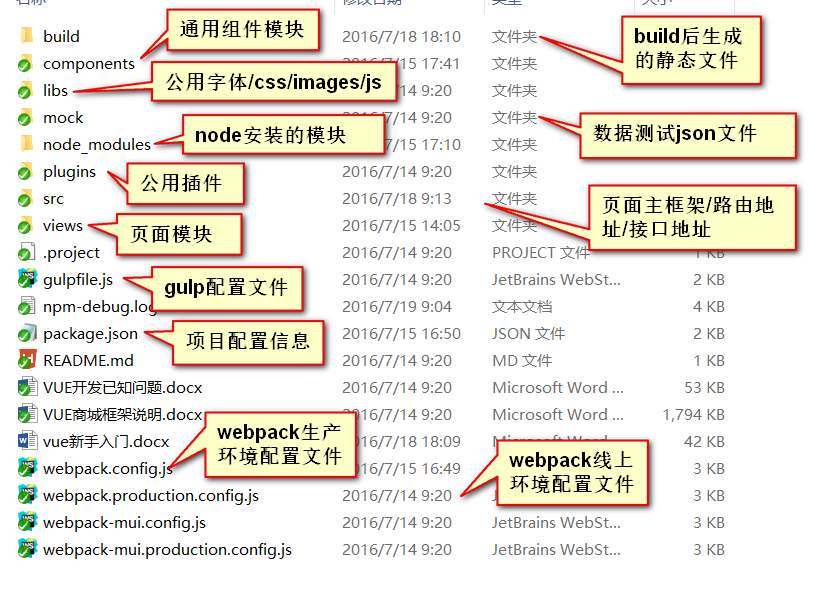
会出现如上图，然后点击”在此处打开命令窗口”，在打开的命令行工具里面输入“npm run dev”安静等待一会命令行会闪过一些花花绿绿的命令，不用做任何操作，等它全部执行完之后，打开浏览器，在浏览器里面输入localhost:8080 即可看到当前项目的首页。

生成静态文件

在命令行里面输入“npm run build”，安静等待一会命令行会闪过一些花花绿绿的命令之后会在项目根目录下生成一个build文件夹，里面的index.html就是主文件，打开即可看到首页，更改hash值可以看到相应的页面。

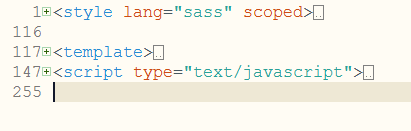
注：这样生成的文件是没有经过压缩和一些优化的，文件会比较大，要生成线上环境的文件请在命令行里输入“gulp”。

项目文件划分图

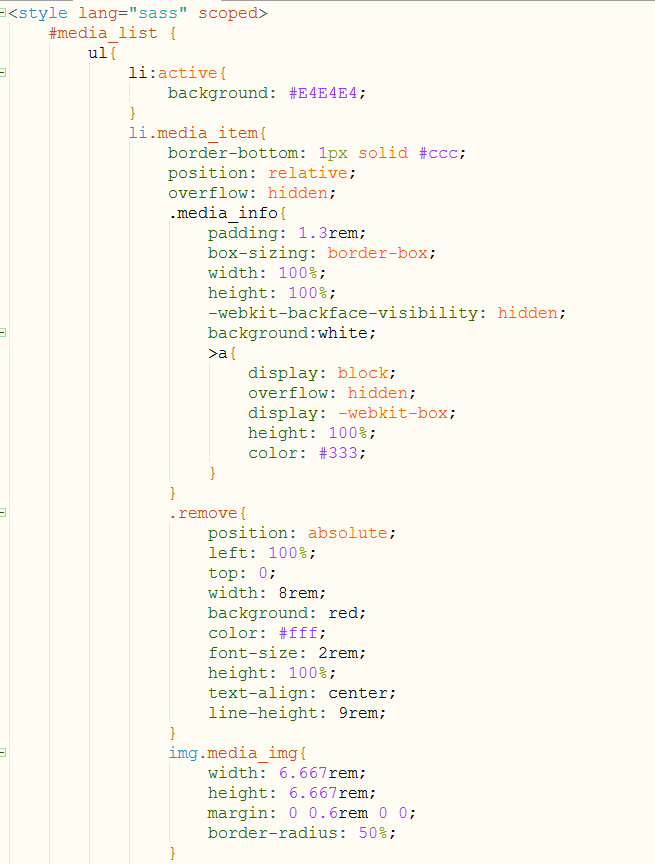


这里关于详细的技术不做讲解，举例说明一下一个页面模块的制作。

1. 在views下面新建一个后缀为vue的文件
2. Vue文件主要分为style/template/script三部分。

如图：

Style：很显然值的就是css样式，当然你也可以使用less/sass css预编译语言，框架里面支持less/scss/sass这三种语法，我这里用的是sass。

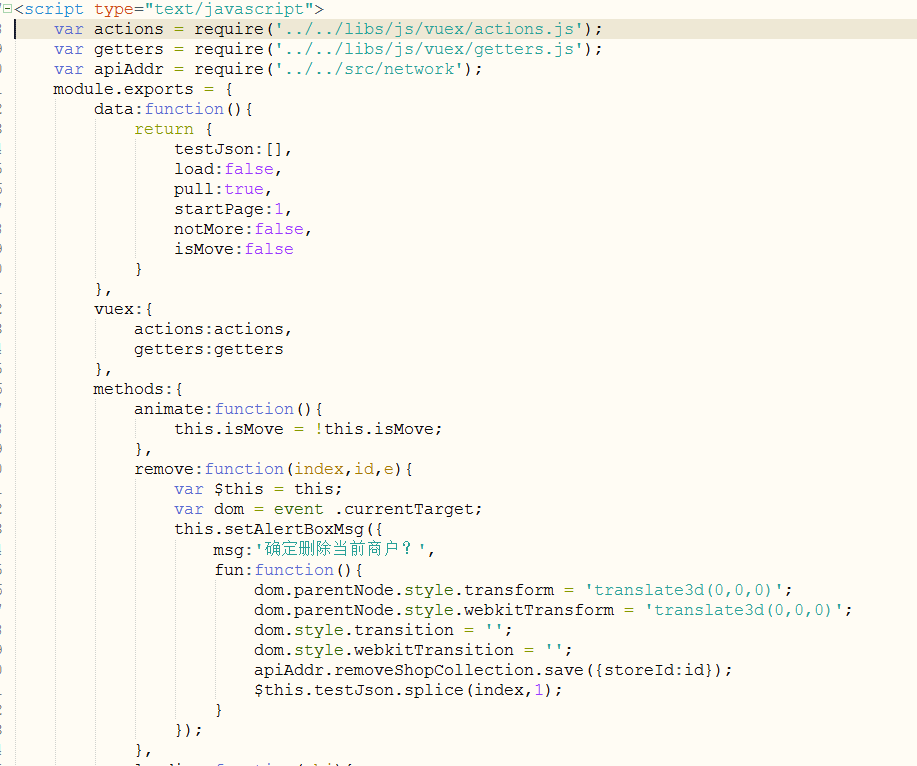


Template：顾名思义，指的就是页面主结构html；



Script：看样子就知道JS嘛。

以上三部分就是一个vue模块的基本内容。



忘了件重要的事，配置路由：

当你做的vue组件想要看效果的话，就打开src/routerConfig.js



根据这个里面的，随便复制一个，然后改hash值，name和模块地址。

注：name不能和上面已有的相同！

页面跳转: <a link={ v-link="{ name: 'user'}"}></a>

Name就是你的页面名字

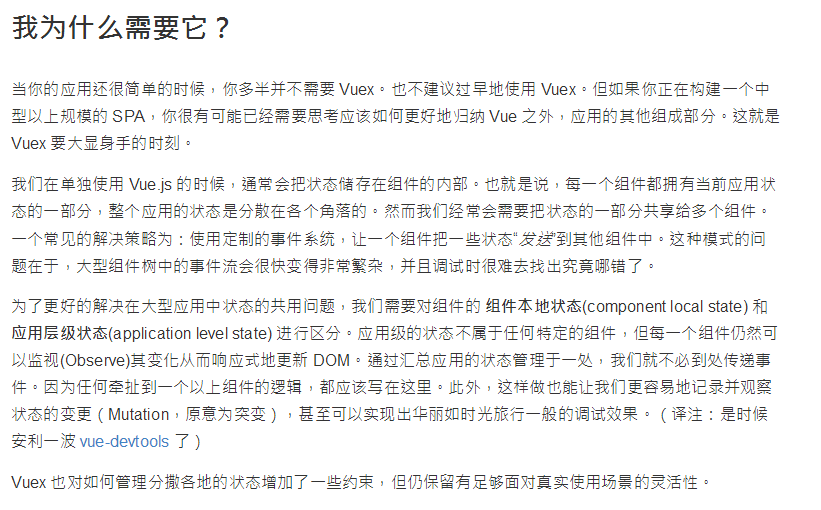
做好上面的步骤之后，这时你可能已经打开了localhost:8080了，没有打开的请先打开，它默认会是<http://localhost:8080/#!/main>，这时你把后面的main改成你自己页面的hash值就能看到你的那个页面了。

**组件通信**

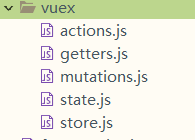
当你有认真的看完vue的官方文档之后，关于组件通信这一块应该也有一个大致的了解了。组件通信：指的是两个或多个不同模块之间可以传递信息并且可以同步更新，vue的实现方式和angular的很像同时也很繁琐。

例：比如我一个vue模块内部有用到，两个不同的组件，而现在这两个组件之间可能要传递一些信息，这样子我们就要在这两个组件之中的其中一个组件使用$dispatch()派发事件，它会沿着父链向上冒泡。然后他们两个的父组件捕获到这个事件，再使用$broadcast() 广播事件，事件向下传导给所有的后代。其中的一个组件接收到这个广播事件之后做一些处理就完成了这两个组件之间这一次的通信。

由于这样做很麻烦，所以我们引入了vuex。



这里对于vuex我也不做过多的介绍，只做我们一个vue模块正常使用的步骤：



这是我们vuex的文件划分

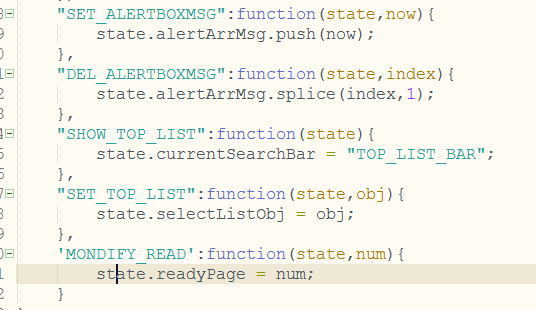
Store.js store 存储应用所需的数据。所有组件都从 store 中读取数据，它会挂载到我们的主框架文件上，这样我们所有的vue组件就都可以通过vuex来完成组件之间的通信了。

State.js 这个文件主要记载了组件默认状态的信息

如图：

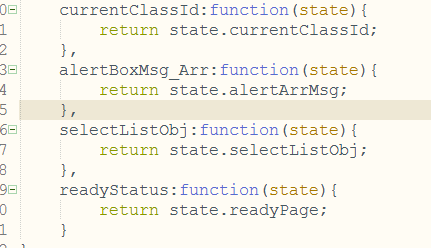
每一个属性代表的都是一个会和其他组件通信的对象。

Mutations.js



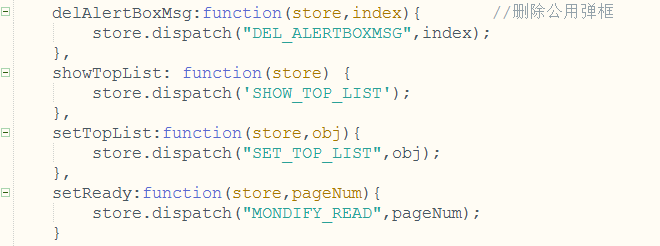
如图，每一个函数代表的是一个更改当前state里面某个属性值的方法，它有一个state参数，通过这个参数可以读写state.js里面的所有属性。

Getters.js



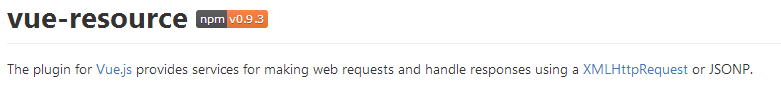
如图，它可以返回state.js里面的任意一个或者多个属性值，让它在页面上面可以获取到。

Actions.js



如图：每一个函数都会有一个事件派发，并且可以传递参数，当actions.js里面的某一个方法触发的时候mutations.js里面的相对应的那个方法就会触发，然后改变state.js里面的属性值，而getters.js就可以获取到state.js里面的属性值，这样就可以方便的完成了组件之间的通信。

讲到这里也就差vue-resource了:



这个是制作者对vue-resource的介绍，意思就是基于vue.js制作的Web请求，提供服务和响应处理，使用XMLHttpRequest或JSONP。

使用方法很简单，我们项目为了便于管理，所以所有页面api都集成在src/network.js里面



如图前面的serverIp都是固定的，后面的就是自己每个不同api的详细地址。

使用方法：

我们常用的方法就是post和get，但是在这个里面post用save代替

如：couponDetail: Vue.resource(serverIp + '/BCYS-Market-biz/card/getCardInstById')；

在要使用的vue页面引入network.js文件



然后

apiAddr. couponDetail.get(obj).then(function(data){

//success

},function(err){

//error

});

上面是get方法，然后看一下post方法的：

apiAddr. couponDetail.save(obj).then(function(data){

//success

},function(err){

//error

});

Get(obj)和save(obj) 这两个括号里面的obj指的就是发送到后台的数据，then里面的两个回调函数第一个是指成功的回调函数，第二个是指失败的回调函数。

讲到这里也就webpack和gulp没有讲了，这两个的话项目已经搭建好了，在制作页面的过程中并不用改动其中的内容，所以这里也就不在过多的介绍，有兴趣的可以去我给出的地址对照着我们项目里面的配置文件看看，还有就是项目里面一些公用的插件和组件都有写说，明，需要的时候可以去看说明。

