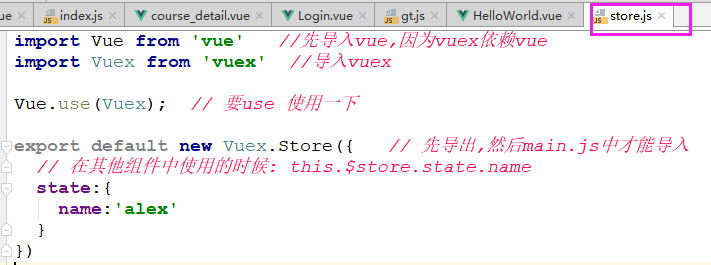
vuex的作用 : 存公用的数据,放在store中

--实例化store对象

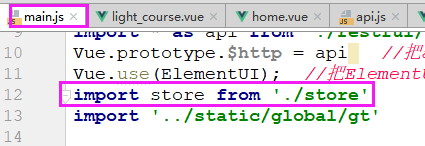
--在vue对象中注册store之后才能用this.$store.state….取值



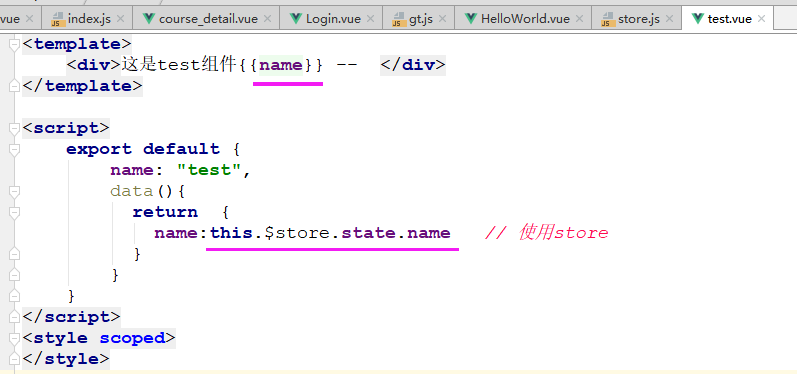
为了解耦单独建一个store.js文件↓:



在main.js中导入这个store.js就可以全局使用这个store了

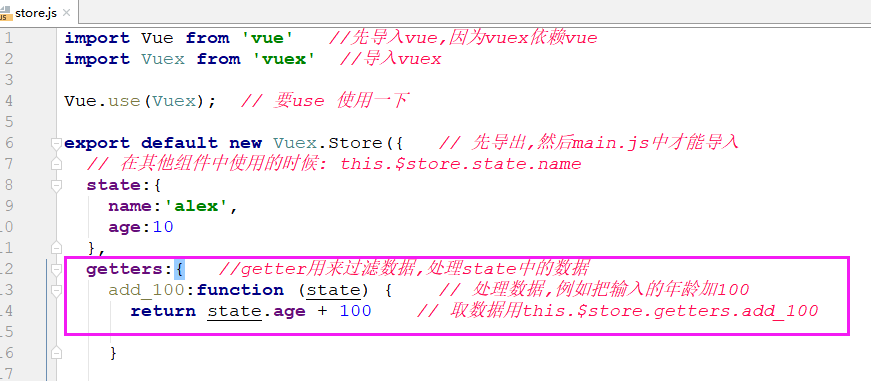


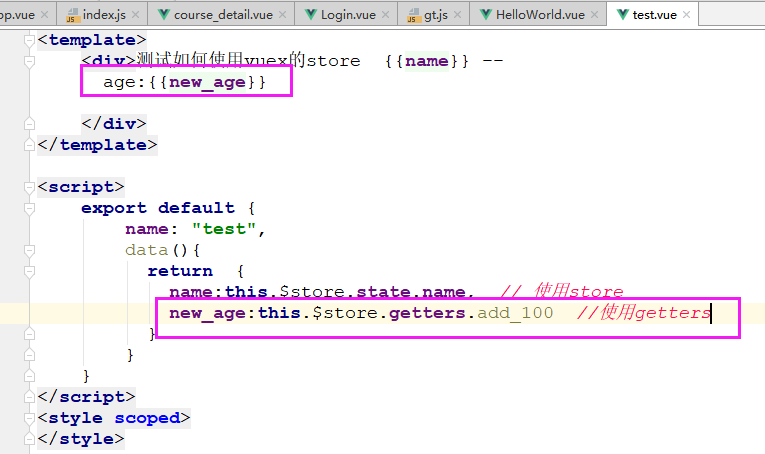
在其他组件中使用store: this.$store.state…..

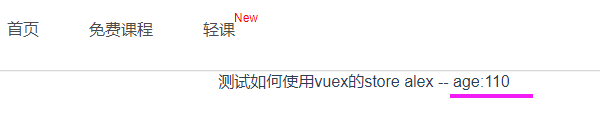


===================================🡺

getters:用来处理state中的数据:

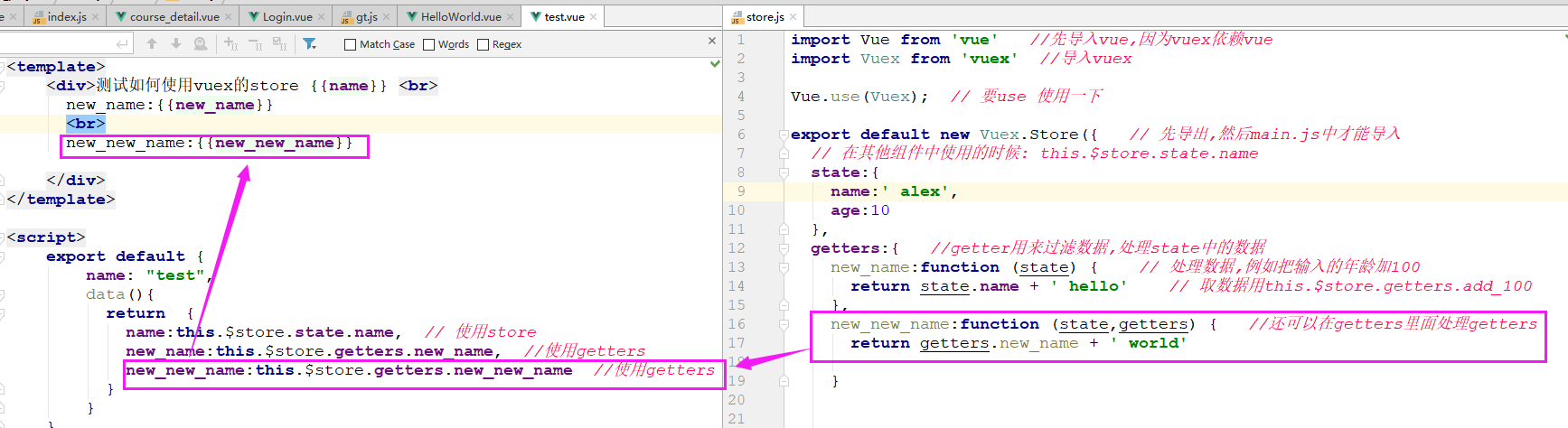


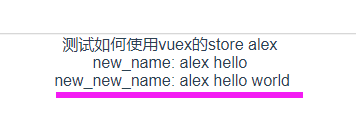




===================================🡺

还可以在getters中处理getters:

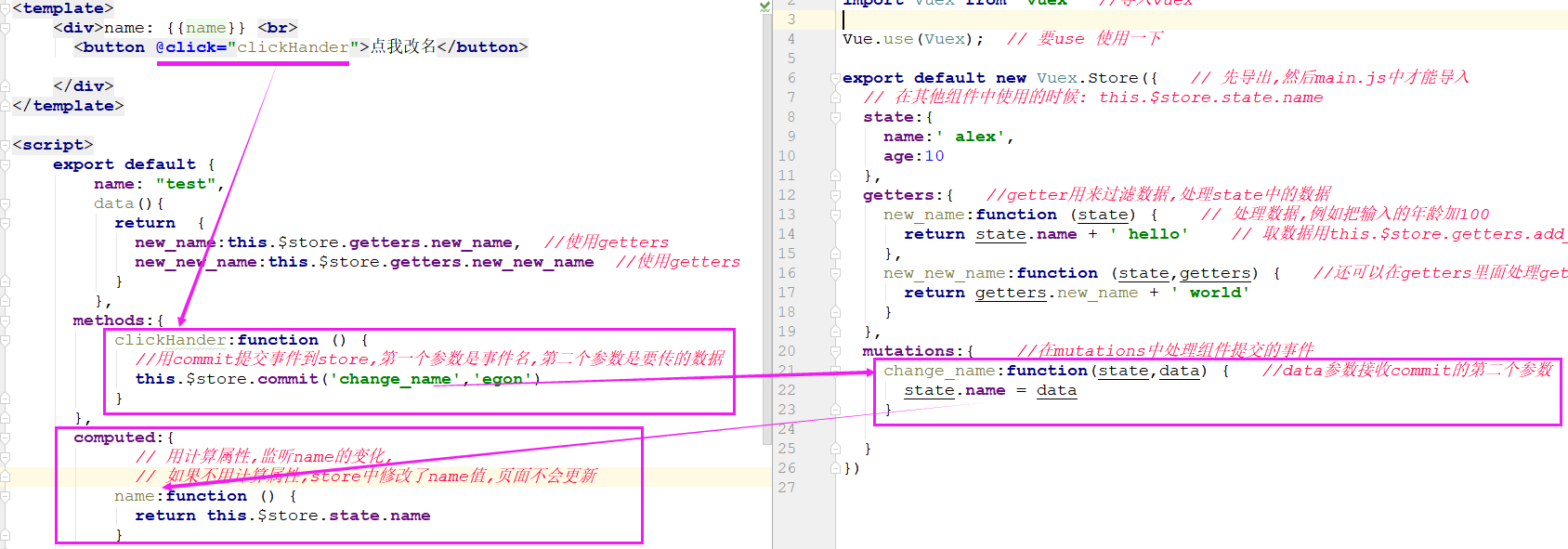




===================================🡺

如果有组件如果要修改数据, 要提交事件通知store, 然后store同步修改其他组件的数据

要用计算属性,不然store修改数据后,页面数据不会更新. 用计算属性computed就会监听数据变化, 所有从store中拿的数据都用计算属性.



==================================🡺

前台发送请求到后台取数据,用js或jq发ajax请求也可以, vue推荐用axis发请求取数据.

vue开放了store根实例属性,所以我们在实例中写store中能识别, 但是vue没开放axios的根实例属性,所以在vue对象中写axios不能识别.如果不用根实例属性,那么每次发请求都要写

import axios from ‘axios’ #这个import后面的名称随便写,这里写成axios

axios.request({

url:xx,

method:'get'

})

很麻烦,所以我们可以这么做:

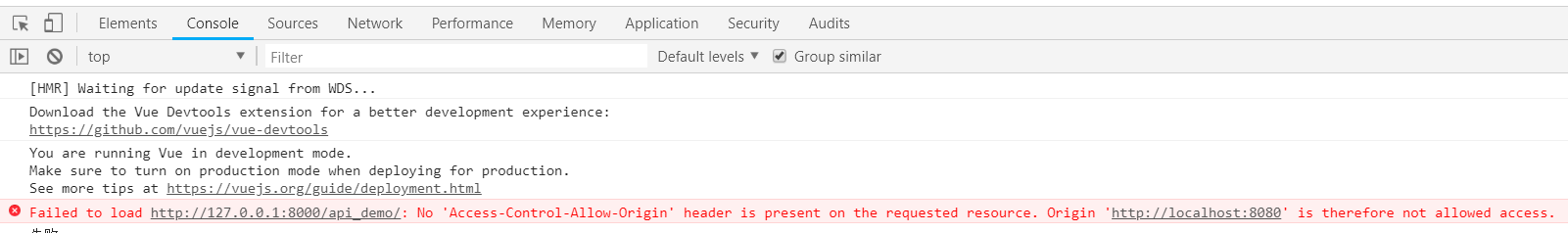
在main.js中添加Vue.prototype.$axios=axios

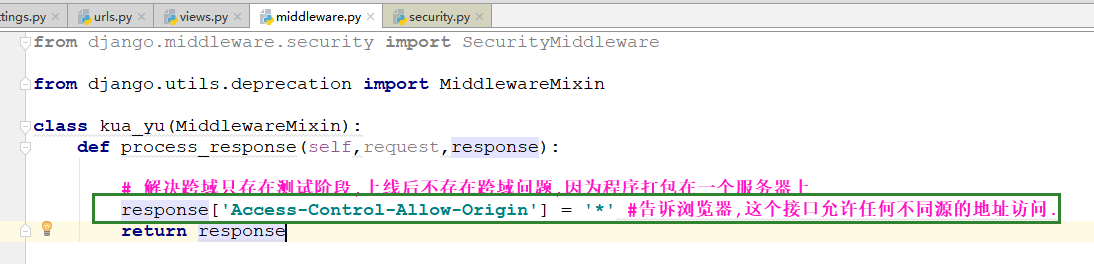
这样以后Vue的对象就有了axios属性.就类似于给python的类加一个属性, 那么类的对象就有了这个属性.

以后就用this.$axios就等于使用axios.

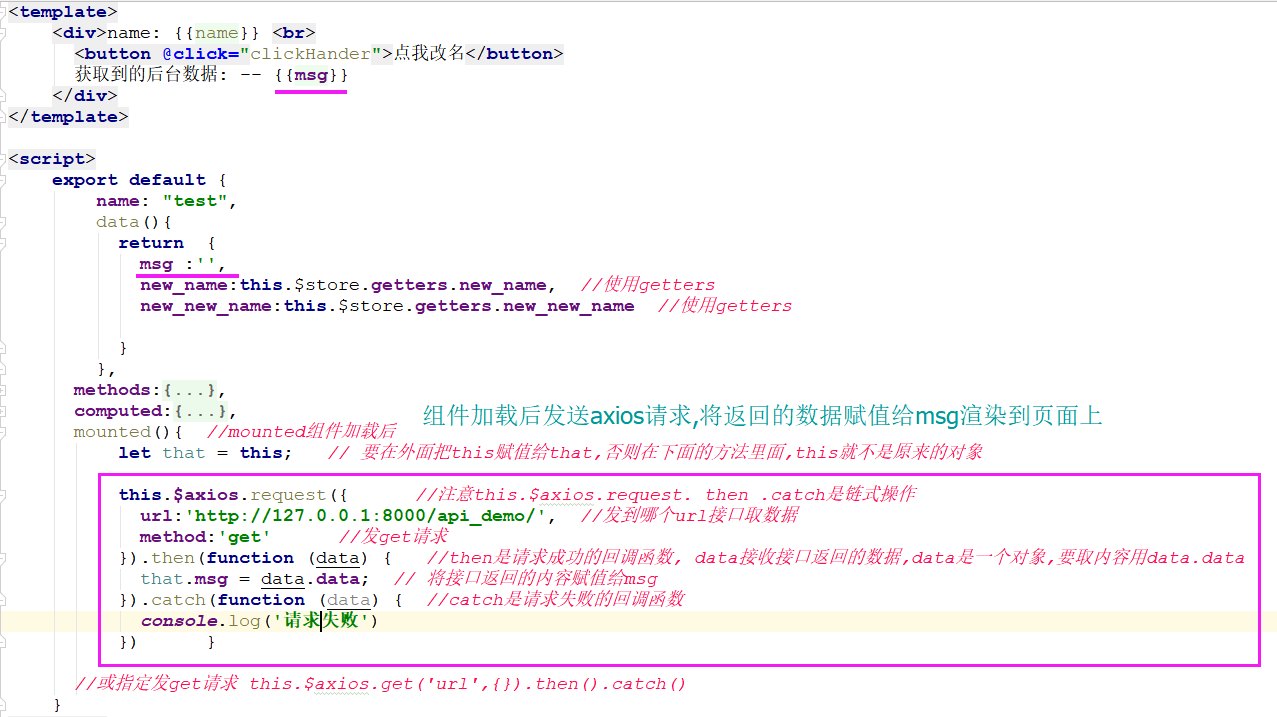


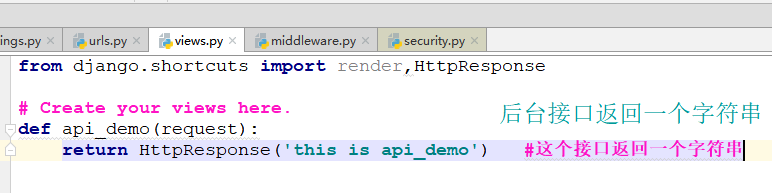
=================================🡺

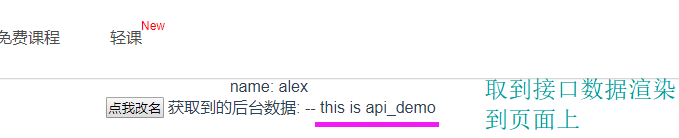
跨域拦截是浏览器拦截的, 浏览器认为不同源的请求有风险所以拦截. 只拦截跨域的ajax请求因为浏览器认为跨域的ajax请求有风险.怎么解决?写一个中间件,缺响应头就加响应头



取接口的数据渲染到页面上↓↓↓:







通过axios发ajax请求向后端获取数据, 然后利用数据驱动视图改变前端页面的显示.\*\*\*\*\*\*

==================================🡺