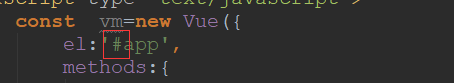
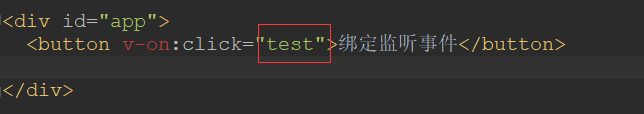
# Vue问题笔记日常记录

## 一、基础篇

1.el:忘了写‘#’.

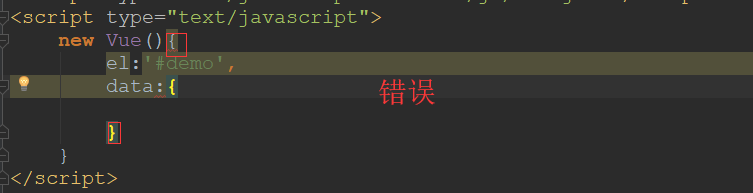


2. 绑定事件监听时，若vm为methods:,则在绑定函数时只需写函数名，不加‘()’.

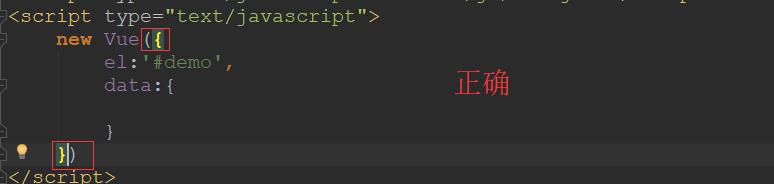


3.创建Vue()实例时，总是将‘{}’写在‘()’外。

错误：



正确：



4.将model错写为modle

5.使用了Vue的修饰符delete 报错

6.div标签没有写id属性

7.ES6中的**字符串拼接**，使用``包裹，传递数据对象使用**$**{Object}

<router-link :to="`/home/message/messageDetail/**$**{message.id}`">

## 二、组件化编码

1.拆css原则

a)、那个组件的放在那个组件上去，

b)、公用的可以创建一个base.css，让main.js引入（此时，所有的组件都可以用）,import ‘base.css’

c）bootstrap.css位置放在index.html

<link rel="stylesheet" href="/static/css/bootstrap.css">

2.批量获取属性

const {todo,index}=this

3.将数组todos的属性complete全部改为ischeck状态（isCheck为true或false）

isAllcheck(ischeck){

this.todos.forEach(todo=>{

todo.complete=ischeck}

)

}

4.过滤器数组，得到状态为true的数据

即删除所有为true的元素，留下为flase的元素

this.todos=this.todos.filter(todo=>!todo.complete)

5.计算数组todos中complete值为true的元素的数量

computed: {

completedCount() {

return this.todos.reduce((preTotal, todo) => preTotal + (todo.complete ? 1 : 0), 0)

}

}

6.模板template的标签必须写在div标签中

7. 配置对象的属性名都是一些确定的名称，不能随便修改

new Vue({//配置对象的属性名都是一些确定的名称，不能随便修改

el: `#app`,

components: {

App

},

template: '<App/>'

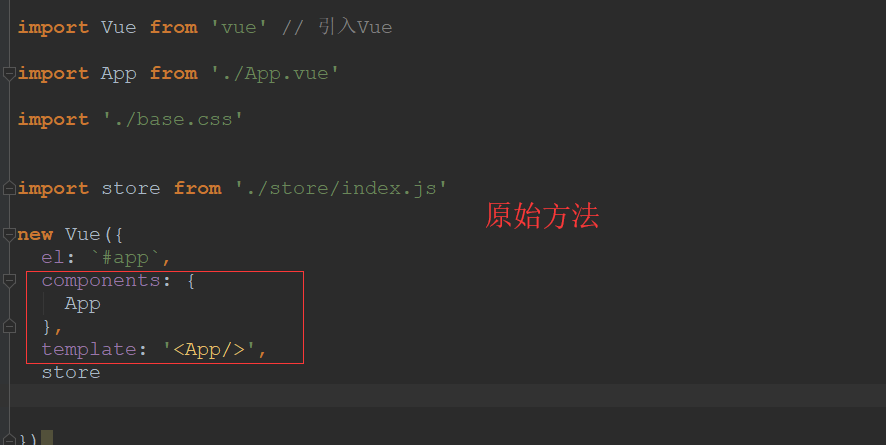
})

8、export default { } 没有括号()

new Vue({ }) 有括号()

9．在main.js中，映射App标签可以有2中方式

方式一：

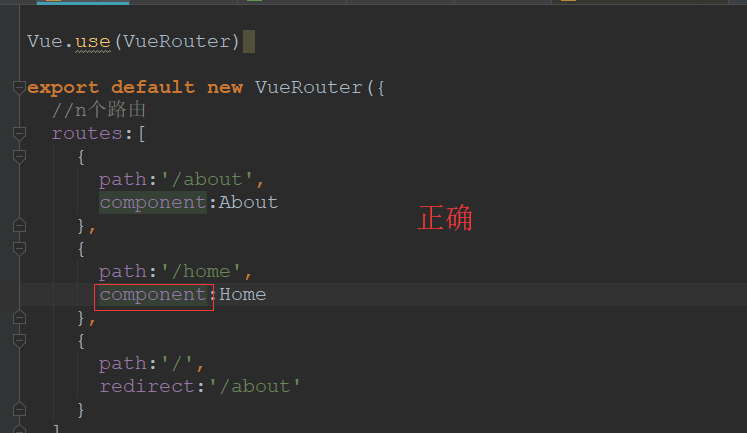


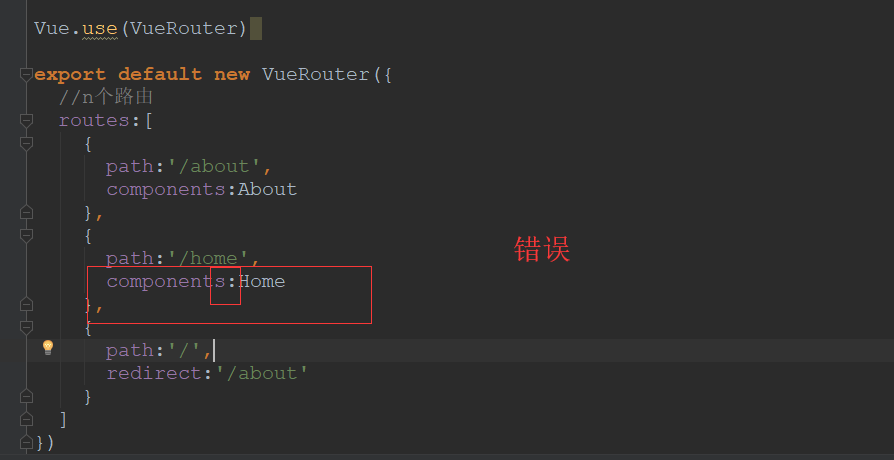
方式二：利用render属性



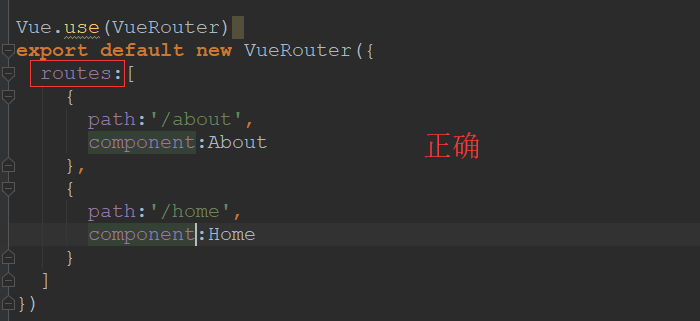
## 三、路由

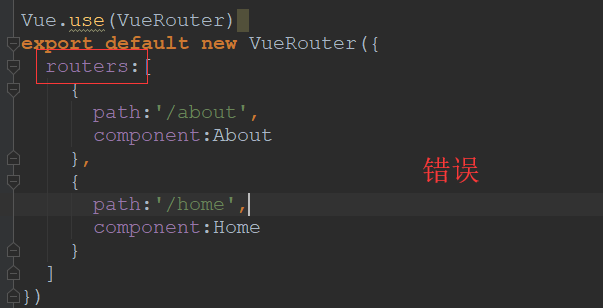
1.在router/index.js中，设计路由时，将componet多加了一个s



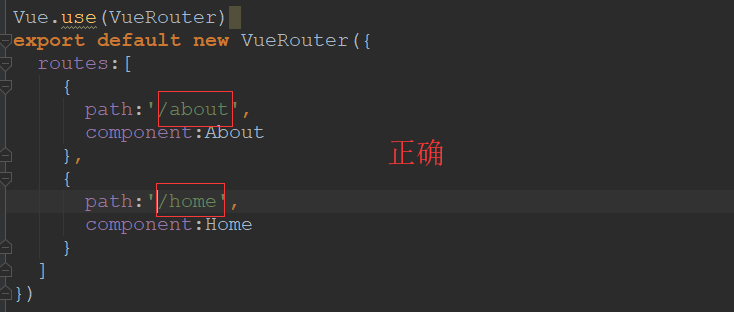


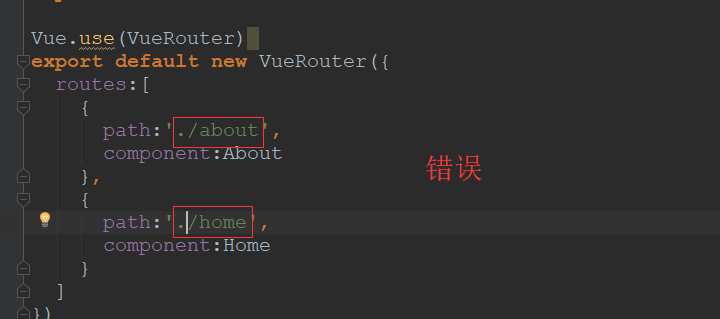
2.在/router/index.js中，将routes错写为routers导致错误





3. 在/router/index.js中，路径书写错误。写成‘./about’,即使调用时写成<router-link to=’ ./abou’>,两边相同也调不到About.vue





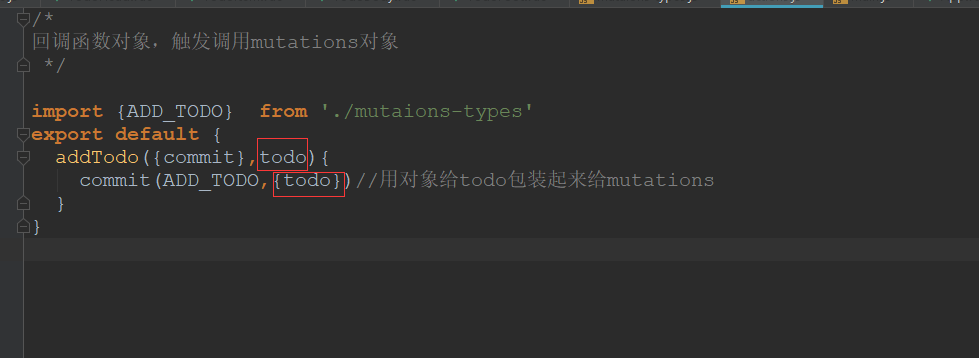
## 四、Vuex

1.忘记在main.js中注册store

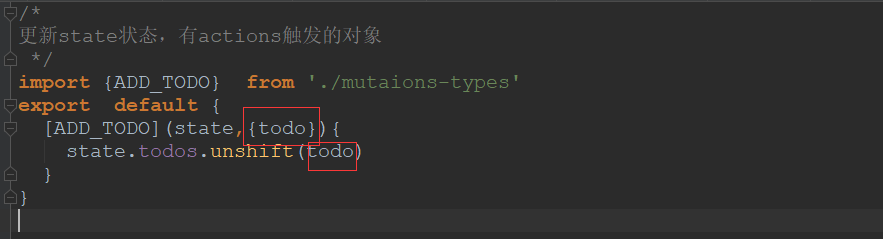


2.在传递参数时，须将其包装为对象，加上{ }包裹即可

在actions.js中传递参数时：

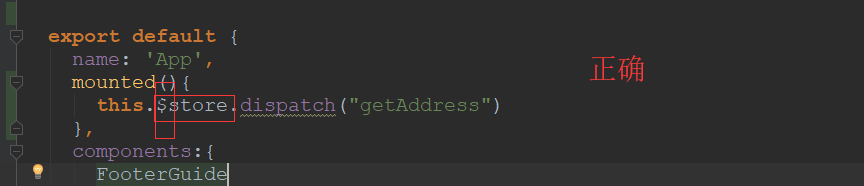


在mutations.js中传递参数时:

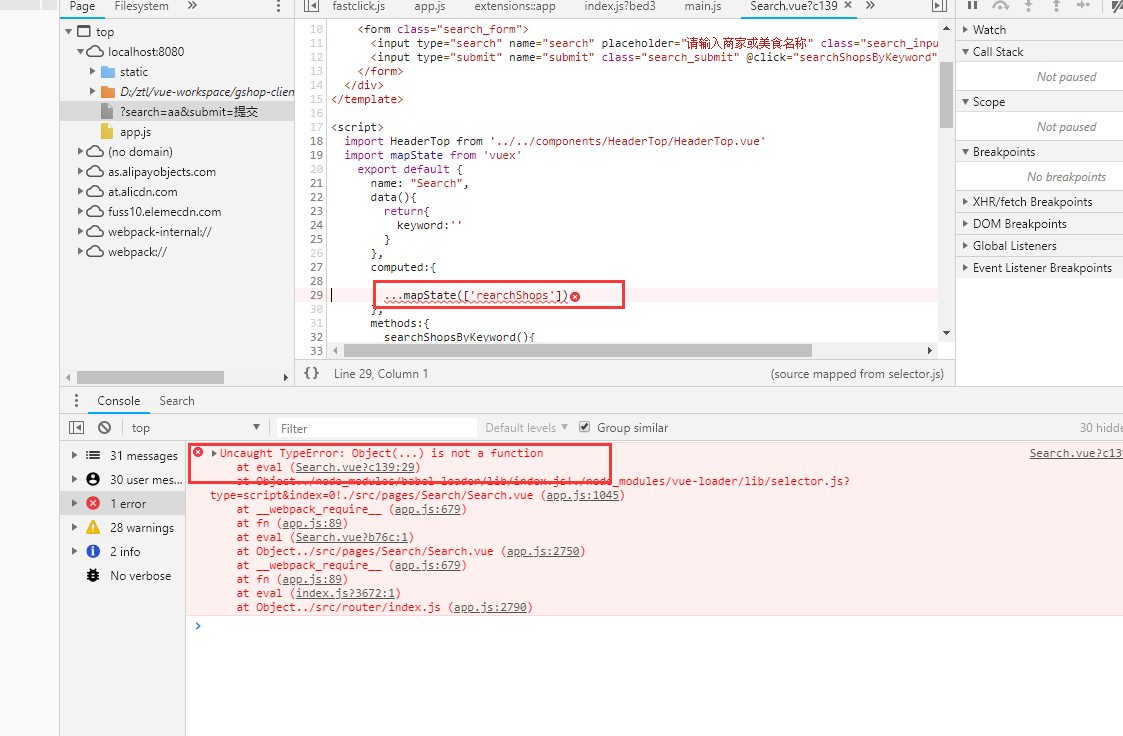


3.在使用store时，必须加$



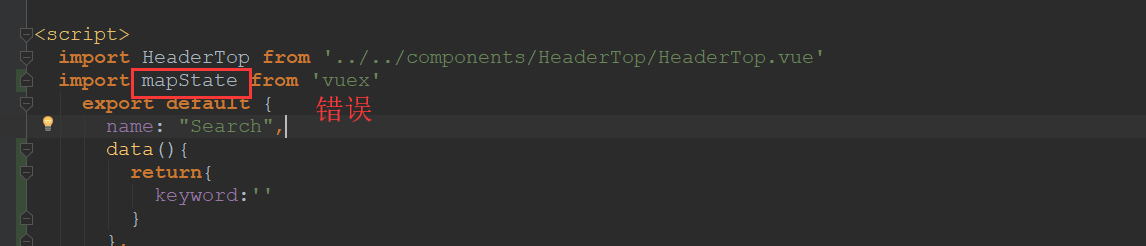


4.在使用mapState时报错，，如下所示：



原因：引入mapState出错

解决：在引入时将mapState用大括号{ }包裹





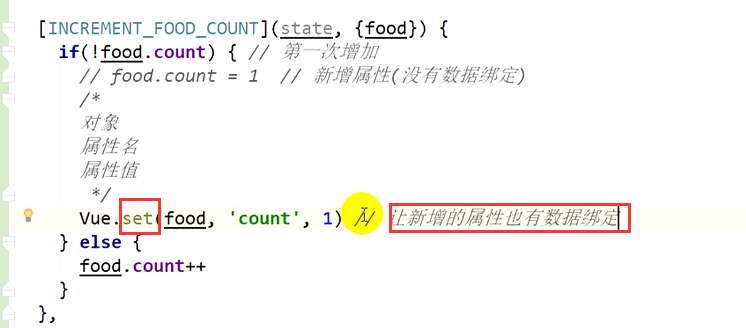
5.问题: 更新状态数据, 对应的界面不变化？

原因: 一般方法给一个已有绑定的对象中添加一个新的属性, 这个属性没有数据绑定

解决:

Vue.set(obj, 'xxx', value)才有数据绑定

this.$set(obj, 'xxx', value)才有数据绑定



## 五、Swiper

1. swiper的创建必须在数据更新到页面之后，仅仅数据更新没用

swiper的滑动效果是在数据已经展现在界面上之后才会有效果，所以swiper的创建是在数据更新到页面之后，假如数据改变了但是没有更新到界面，此时创建是没有用的。

此时，使用**this.$nextTick(()=>{})**,界面更新就立即调用,

并且使用watach来监视数据的改变

示例如下：

watch: {

categorys(){  
 **this**.$nextTick(()=>{//界面更新就立即调用  
 //创建Swiper实例对象，来实现轮播  
 **new** Swiper('.swiper-container', {  
 //direction: 'vertical',  
 loop: **true**,  
 // 如果需要分页器  
 pagination: {  
 el: '.swiper-pagination',  
 },  
 })  
 })  
 }  
},

## 六、Ajax前后台交互相关问题

1. 前后台交互相关问题

1). 如何查看你的应用是否发送某个ajax请求?

浏览器的network

2). 发ajax请求404

请求的路径的对

代理是否生效(配置和重启)

服务器应用是否运行

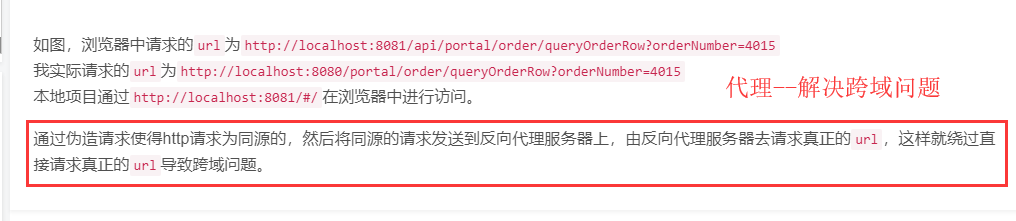
代理是否生效(配置和重启)：相关代码如下

config/index.js 目录

proxyTable: {

'/api': { // 匹配所有以 '/api'开头的请求路径  
 target: 'http://localhost:4000', // 代理目标的基础路径  
 changeOrigin: **true**, // 支持跨域  
 pathRewrite: {// 重写路径: 去掉路径中开头的'/api'  
 '^/api': ''

}  
 }  
},



3). 后台返回了数据, 但页面没有显示?

vuex中是否有

组件中是否读取

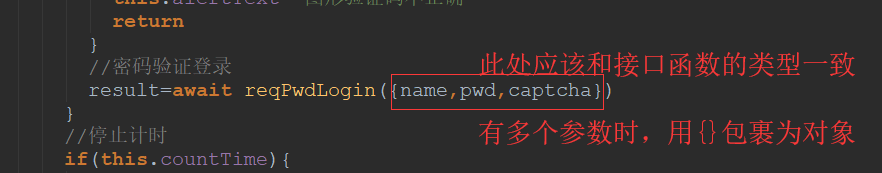
2. 初始显示异常

情况1: Cannot read property 'xxx' of undefined"

原因: 初始值是空对象, 内部没有数据, 而模块中直接显示3层表达式(a.b.c),2层表达式不会报错

解决: 使用v-if指令(不能用v-show，控制显示的方式不同，一个销毁，一个disable)

情况2: Cannot read property 'xxx' of null"

3. 使用接口函数时，注意参数类型一致。多个参数使用大括号{}包裹为对象

## 七、Element-UI

1、下拉菜单Value显示成了Label的值。

for循环枚举，得到所有枚举code和messsage,并填充下拉菜单。

出现问题：获取不到下拉菜单的value,为空或者为label显示的值。

处理：

1)枚举常量如：basicEnum.js

**const** platformType =

{  
 1:'内部用户 ',  
 2:'外部用户',  
 3:'其他用户'  
}  
**export default** {  
 platformType  
}

2)在index.js文件(路由下或者其他js文件)引入，方便在任何地方可以使用

**import** basicEnum **from** "@/constant/basicEnum"

Vue.prototype.BASICENUM = basicEnum

3）获取并填充下拉菜单 xxx.vue

<**el-select** v-model="query.platformType"

size="mini" style="width: 150px" clearable>  
 <**el-option** label="全部" value=""></**el-option**>  
 <**el-option** v-for="(***item*** ,***key***) in platformTypes"  
 :key=" ***key*** "  
 :value=" ***key*** "  
 :label="***item***">  
 </**el-option**>  
</**el-select**>

data () {

**return** {

platformTypes:[] ,//平台类型集合

}

}

mounted () {

**this**.platformTypes=**this**.BASICENUM.platformType**}**

注意：

1. 这里的item应该指的是枚举code, platformTypes[item]指的是对应的message,但是似乎不对。以至于记录了这个问题。由第2点来处理。
2. 在v-model双向获取下拉菜单value时（数字，不是lable文字），value的值用platformTypes[item]或者item都是错误，此时需要使用***key***来处理，这里的***key***以为是当前项的索引，这里表示枚举常量键名。

此问题可参考对应的vue文档，v-for有具体的说明：

[https://cn.vuejs.org/v2/guide/list.html#%E7%94%A8-v-for-%E6%8A%8A%E4%B8%80%E4%B8%AA%E6%95%B0%E7%BB%84%E5%AF%B9%E5%BA%94%E4%B8%BA%E4%B8%80%E7%BB%84%E5%85%83%E7%B4%A0](https://cn.vuejs.org/v2/guide/list.html%23%E7%94%A8-v-for-%E6%8A%8A%E4%B8%80%E4%B8%AA%E6%95%B0%E7%BB%84%E5%AF%B9%E5%BA%94%E4%B8%BA%E4%B8%80%E7%BB%84%E5%85%83%E7%B4%A0)

2、下拉菜单默认值问题

**问题描述**：

在表单修改页面中，下拉项的值应该有对应的默认项。而结果是下拉菜单默认选项是value值。

**期望**：默认选项应当是label值。

**原因**：一般来说，有2中情况。其一是v-model与:value的数据类型不匹配造成。另外是：列表数据(下拉项集合)在详情数据(包含默认项)之后加载；简单来说就是：遍历数据和赋值的先后顺序，必须保证数据先遍历后赋值。这种情况可能是异步获取数据导致数据遍历数据在赋值之后。

**实战原因记录**：类型不匹配导致。将:value的数据类型转为与v-modle一致。

**Vue中类型转换**：

数字转字符串：value.toString();

字符串转数字：parseInt(value);

参考链接：<https://blog.csdn.net/carmen_shao/article/details/82107679>