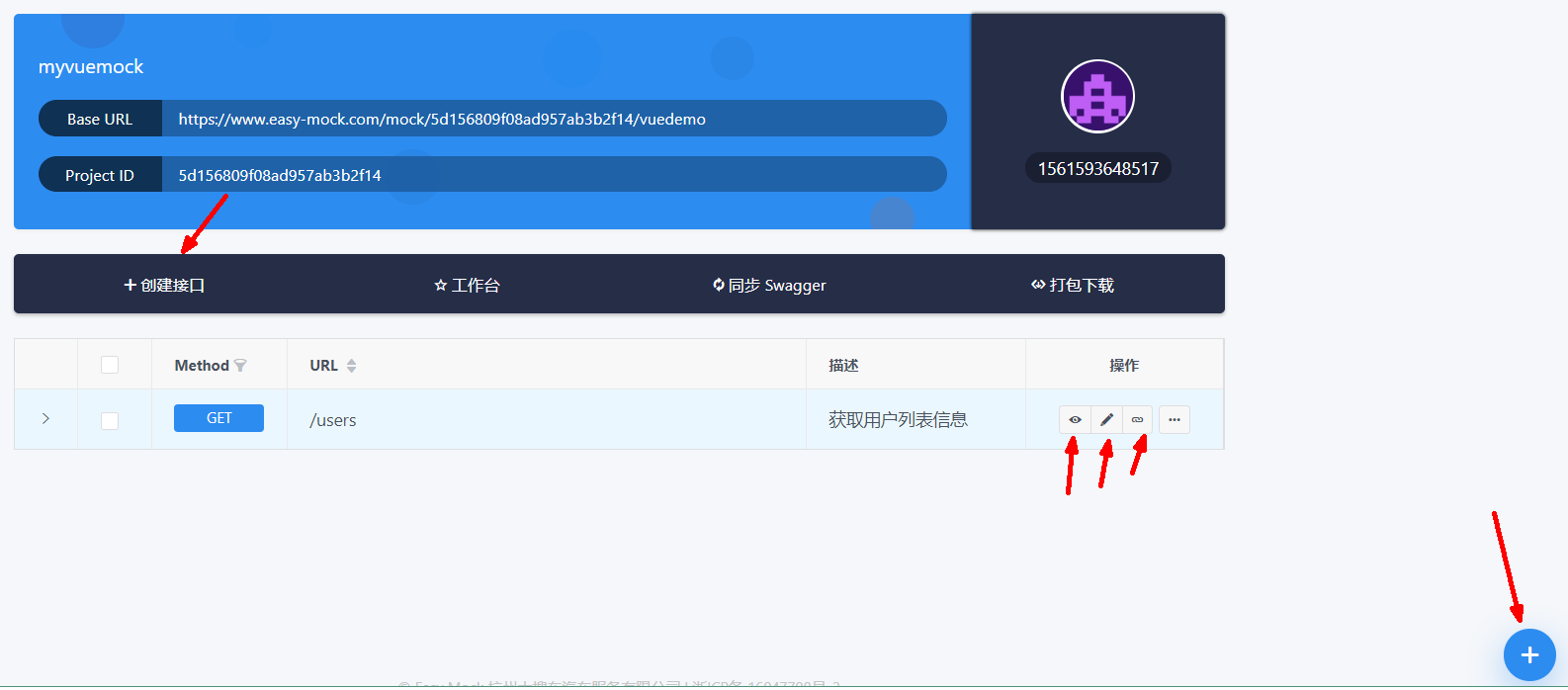
# 一、Mock数据

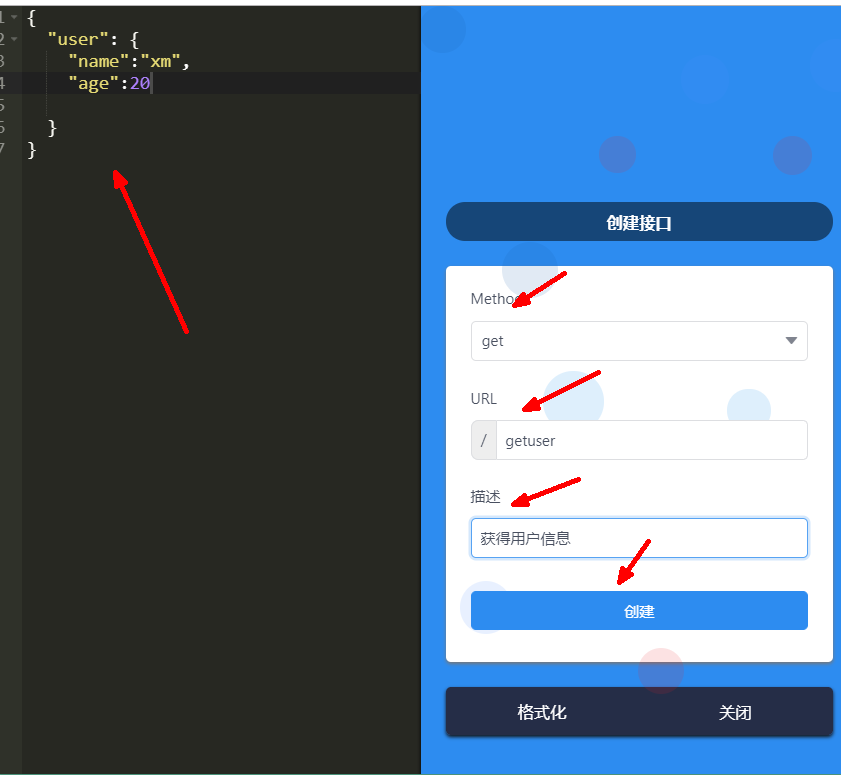
假数据，提供假数据，可以让前端不需要等待后端的接口，而直接进行下一步的开发。

一个常用的工具： easy mock

## 1.进入到easymock官网去在线创建接口

[https://www.easy-mock.com](https://www.easy-mock.com/)





## **2.在发送axios请求时，如何调用到当前vue对象中的内容**

***因为axios内部的then内的this是当前axios对象，而不是当前vue对象，因此把当前vue对象用vm来指明。这样this不会冲突***

|  |
| --- |
| created(){  **var** vm = **this**;   **this**.**axios**({  **method**:**'get'**,  **url**:**'https://www.easy-mock.com/mock/5d156809f08ad957ab3b2f14/vuedemo/users'** })  .then(**function**(resp){   }); } |

# 二、嵌套路由（子路由）

在路由显示的组件内部，又嵌套着路由，称为子路由。

嵌套路由的实现：

## 1.配置路由表

children

|  |
| --- |
| *// 配置路由* **export default new** Router({  **routes**: [  {  *// 路由路径* **path**: **'/Login'**,  *// 路由名称* **name**: **'Login'**,  *// 跳转到组件* **component**: Login  },  {  **path**:**'/Home'**,  **name**:**'Home'**,  **component**:Home,  **children**:[  {  **path**:**'/productlist'**, //注意，使用路由时的地址：/productlist  **name**:**'ProductList'**,  **component**:ProductList  },  {  **path**:**'/productinfo'**,  **name**:**'ProductInfo'**,  **component**:ProductInfo  }   ]  }  ] }); |

## 2.使用嵌套路由

<**router-link to="/productlist"**>商品列表</**router-link**>

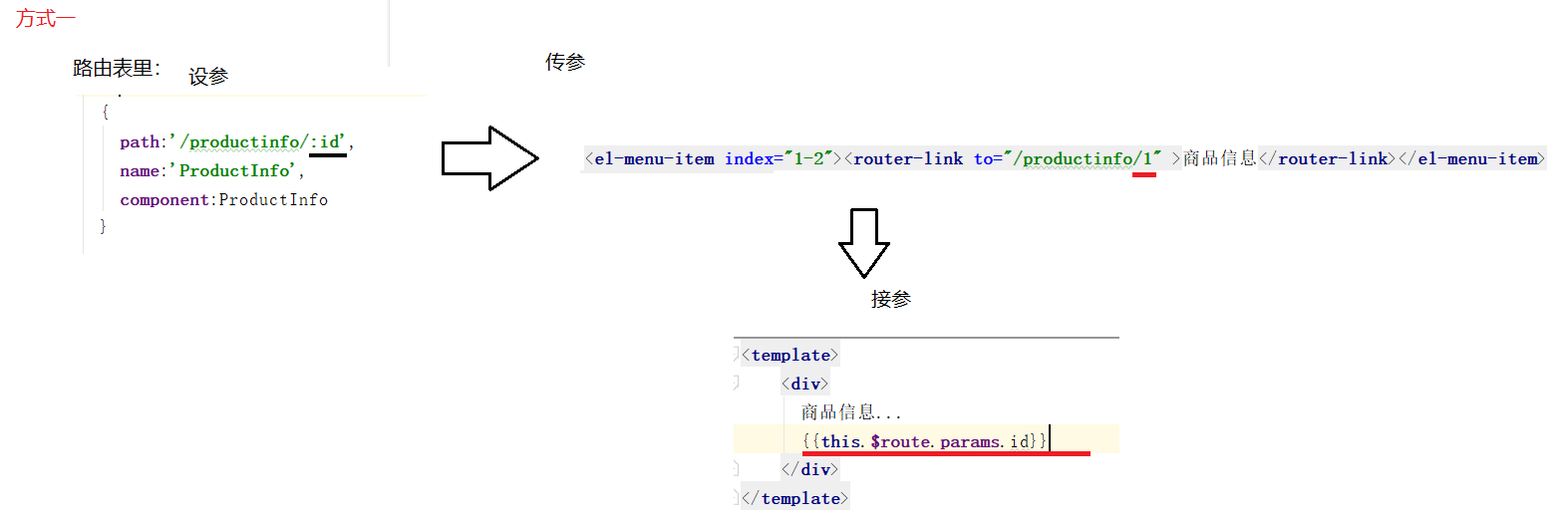
# 三、关于路由的一些细节

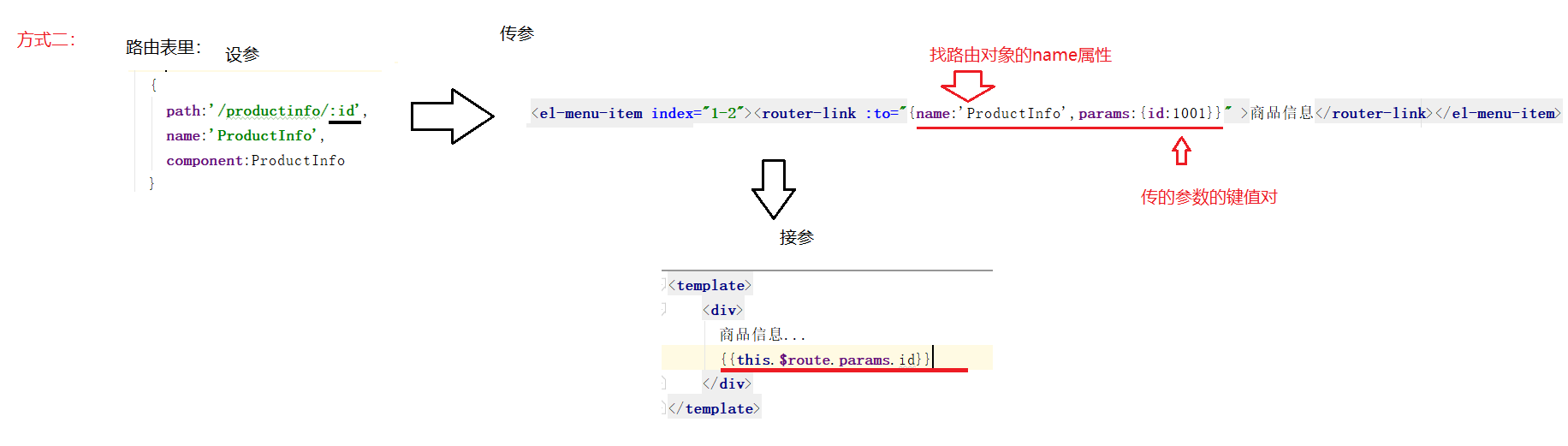
## 1.路由的重定向

直接调用另一个已配置好的路由对象即可。

|  |
| --- |
| **routes**: [  {  *// 路由路径* **path**: **'/Login'**,  *// 路由名称* **name**: **'Login'**,  *// 跳转到组件* **component**: Login  },  {  **path**: **'/Logout'**,  **redirect**: **'/Login'** }, |

## 2.传参的两种方式



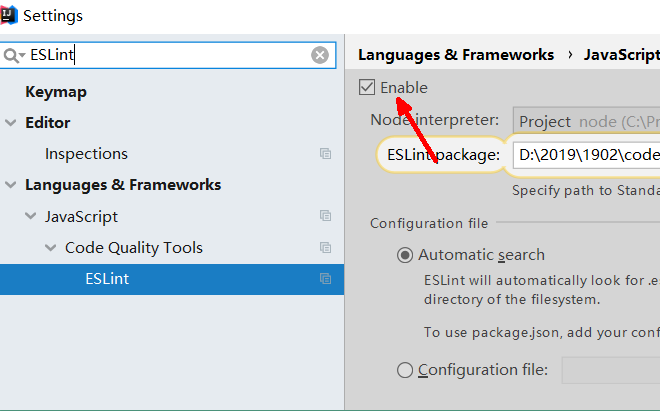


# 四、vue-element-admin后台系统的项目结构

## 1.注意点：

idea在使用vue-element-admin 模板时会默认开启ESLint代码质量管理工具。

为了开发方便，关闭ESLint。



## 2.路由和侧边栏

整个项目的侧边栏是通过路由表动态生成的，可以调整路由表中的内容来改变侧边栏的内容。

改变侧边栏的标题

{  
 **path**: **"/example"**,  
 **alwaysShow**: **true**,  
 **component**: Layout,  
 **redirect**: **'/example/table'**,  
 **name**: **'Example'**,  
 **meta**: { **title**: **'商品管理'**, **icon**: **'example'** },

# 五、编写后端接口

## 1.获得所有商品的集合

## 2.编辑商品

在商品列表中提供编辑按钮

创建一个商品编辑页（editproduct.vue）

在路由表中注册商品编辑页

通过商品列表的编辑按钮跳转到商品编辑页

解决第一个问题：

路由时传递商品id的参数，通过之前的套路即可解决

解决第二个问题：

通过id，发送请求，让后端提供该id的商品对象，并展示在页面中

解决第三个问题

点编辑按钮时，获得所有的数据，传送给后端接口，实现修改数据的效果

请求方式是用POST， 编辑完后，要跳转到商品列表

解决第四个问题：

post请求如何传递数据。

需要使用qs内置库

|  |
| --- |
| import Qs from 'qs'  onSubmit(){  **var** vm = **this**;  **this**.**axios**({  **method**: **'POST'**,  **url**: **''**,  *//=========解决POST请求无法携带数据的问题===========* **transformRequest**: [**function** (data) {  **return** Qs.stringify(data)  }],  *//===============================* **data**:vm.**product** }).then(); } |

## 3.删除商品

## 4.添加商品