

# 购物车列表渲染

作者: [ChenforCode](#)

原文链接: <https://ld246.com/article/1572013376906>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.将mock文件夹放入public文件夹，模仿后台，这个时候可以通过/mock/cart.json的方式访问到相关的json文件，但是要注意把浏览器地址栏的#删掉，要不然是访问不到的。

2.用axios访问后台拿数据，因为现在没有后台，就用mock来模拟一个。这里为什么安装vue-axios个插件呢。首先如果不用这个插件，那么就需要在使用axios的时候提前导入，例如

```
import axios from "axios"
```

然后通过，axios.get()，里边里边的参数为访问地址，例如https://api/v1/getlist

但是如果采用了vue-axios这个组件，那么vue会把axios封装成为一个vue对象，那么通过this就可以接访问的到，不需要再引入axios这个组件了。如下方式

```
this.axios.get();
```

3.如何获取axios返回的数据，如：

这里是第一种方式，这样的嵌套函数是没有作用域的，所以在函数内部也可以使用this，如果没有这，而采取的第二种方式

```
this.axios.get("/mock/cart.json").then((response) => {  
    window.console.log(response);  
});
```

这种方式会有作用域的概念，即在函数中就不能使用this了，这个地方可以类比与微信小程序中，必在外部将this指针保存成一个that才能在内部使用。

```
this.axios.get("/mock/cart.json").then(function(response){  
    window.console.log(response);  
});
```

取出里边的数据基本上是和小程序一致的，

```
this.axios.get("/mock/cart.json").then((response) => {  
    window.console.log(response);  
    let res = response.data;  
    this.cartList = res.data;  
});
```

4.列表渲染

v-for指令，写到要循环渲染的那个标签上，item in cartList，那么item就是那个数组里的一个元素绑定src属性：

```

```

绑定文字：

```
<div class="item-name">{{item.productName}}</div>
```

内部赋值计算

```
<div class="item-price-total"> ¥ {{item.productNum*item.productPrice}}元</div>
```

绑定class

```
<a href="javascript:;" class="checkbox-btn item-check-btn" v-bind:class="{ 'checked': item.chec
```

```
ed}">
```

v-for指令后加上id，提高渲染效率

```
<li v-for="item in cartList" v-bind:key="item.productId">
```