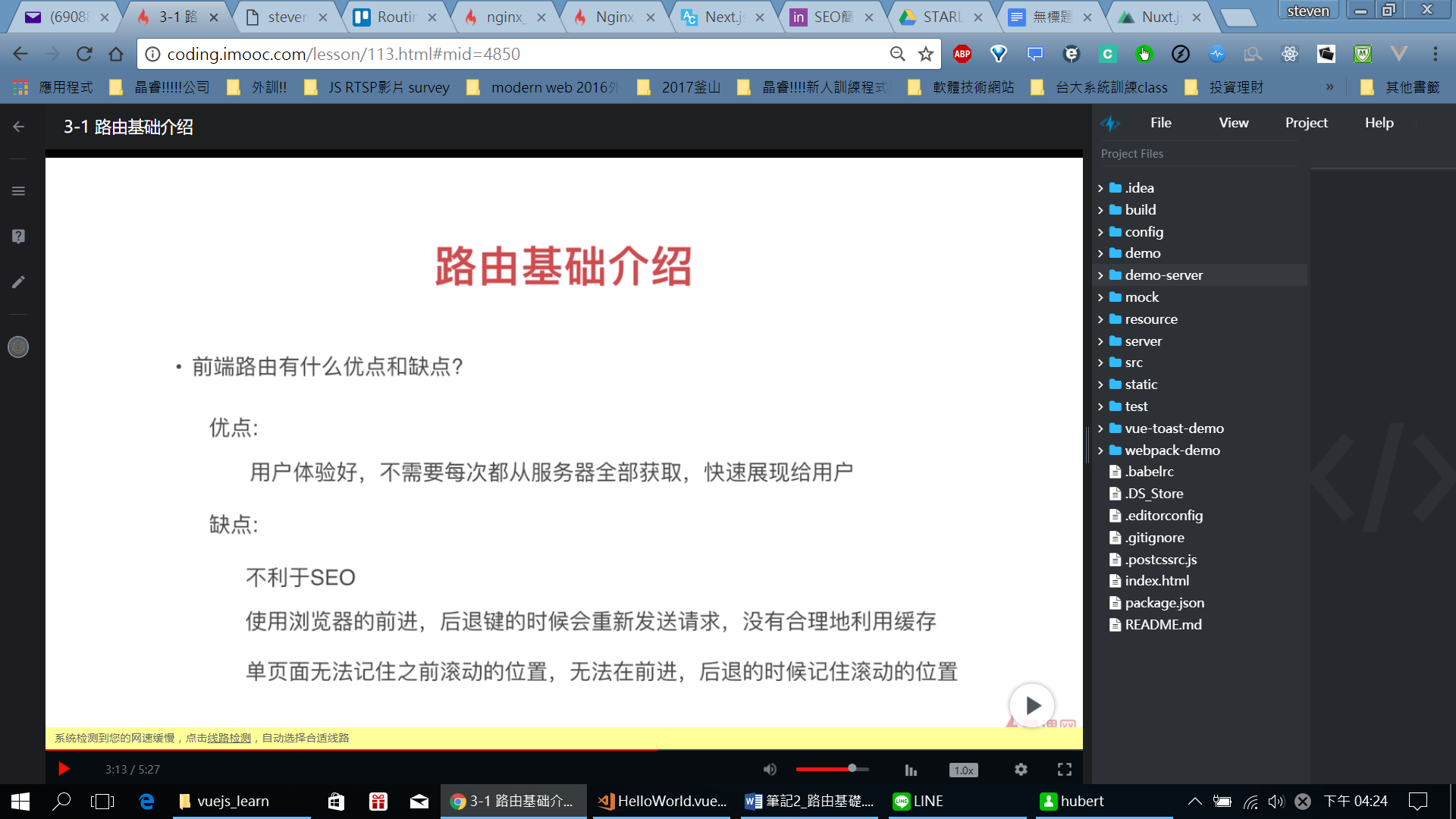
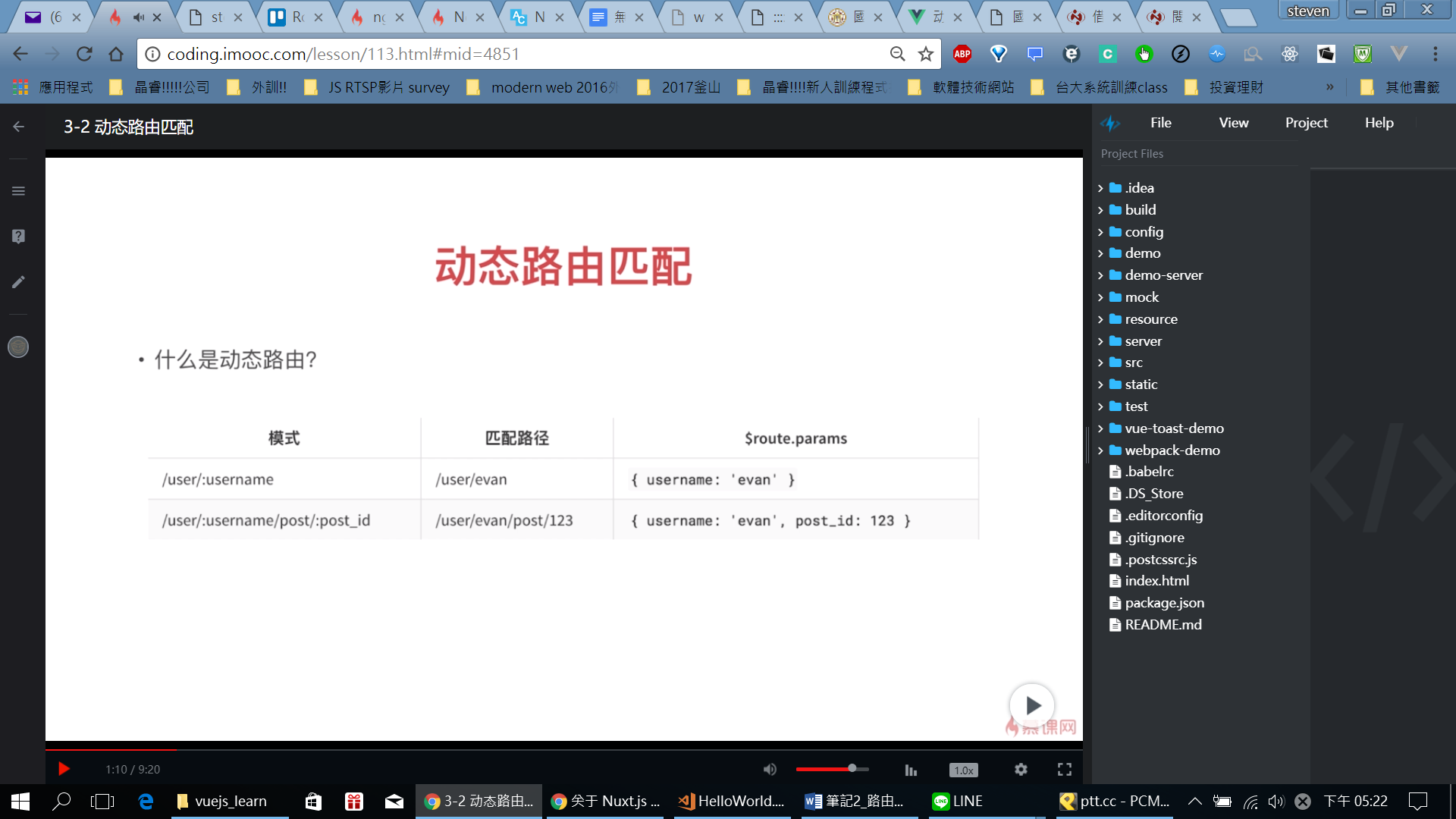
前端路由優缺點:



因為是SP (single page)，可能在google 搜尋時會只搜到第一頁(所以不利SEO搜尋引擎最佳化)

<3-2 動態路由>

2. 動態路由:



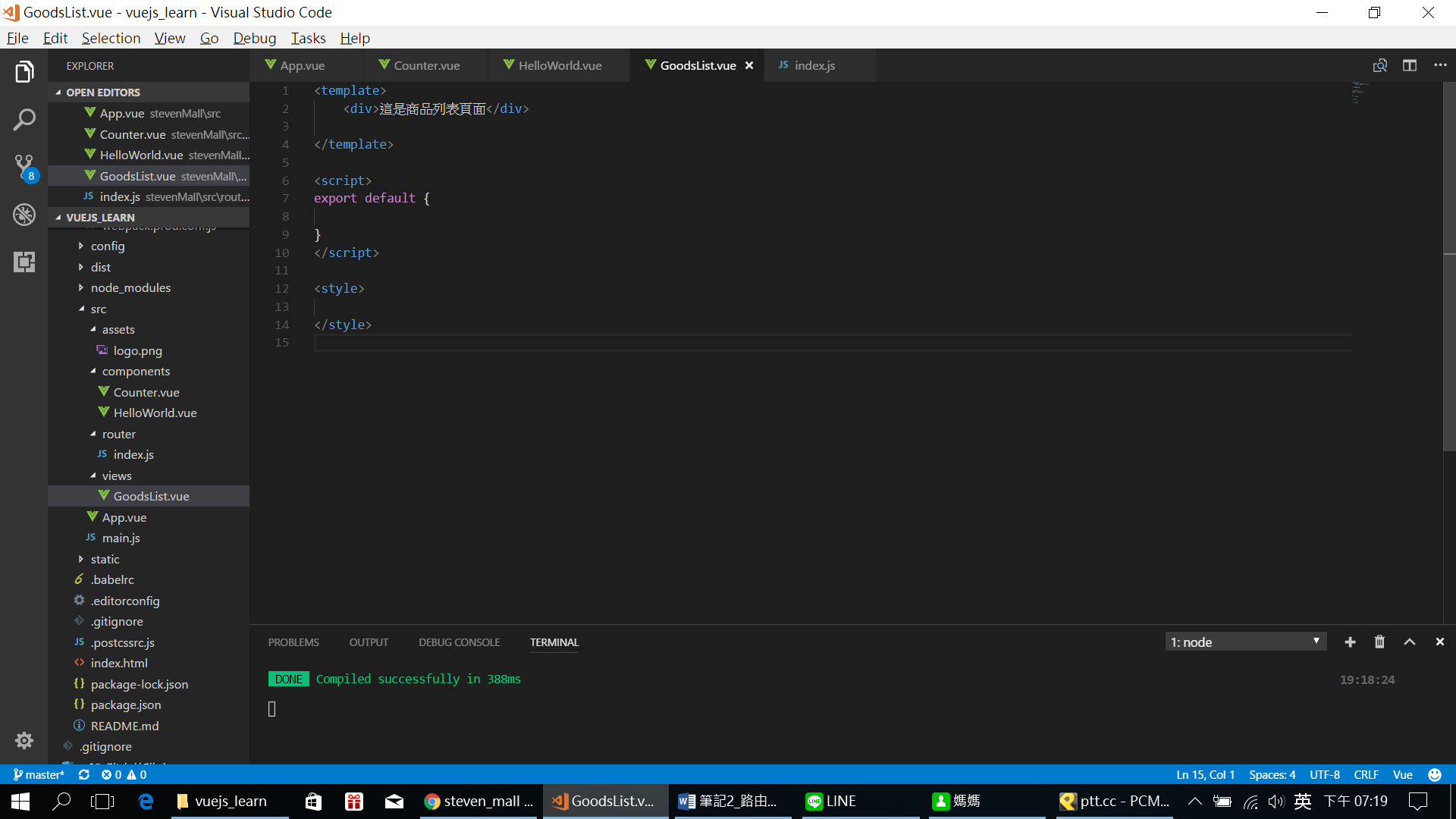
<可以看到路由的長相，並知道$route.params可以拿到什麼>

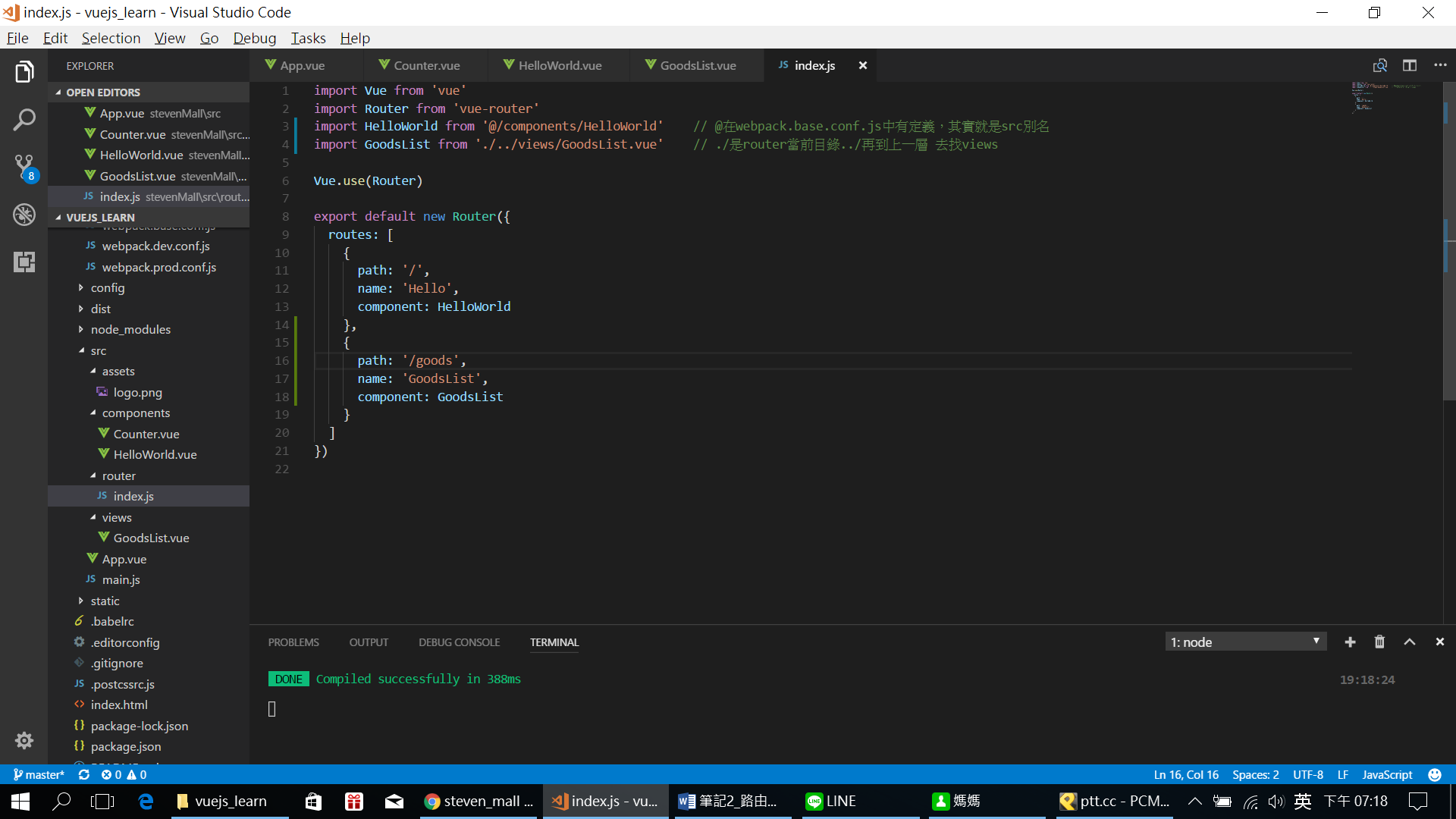
<通常UI端就可以這樣顯示: $routes.params.username 之類的顯示username>

3. 試做:

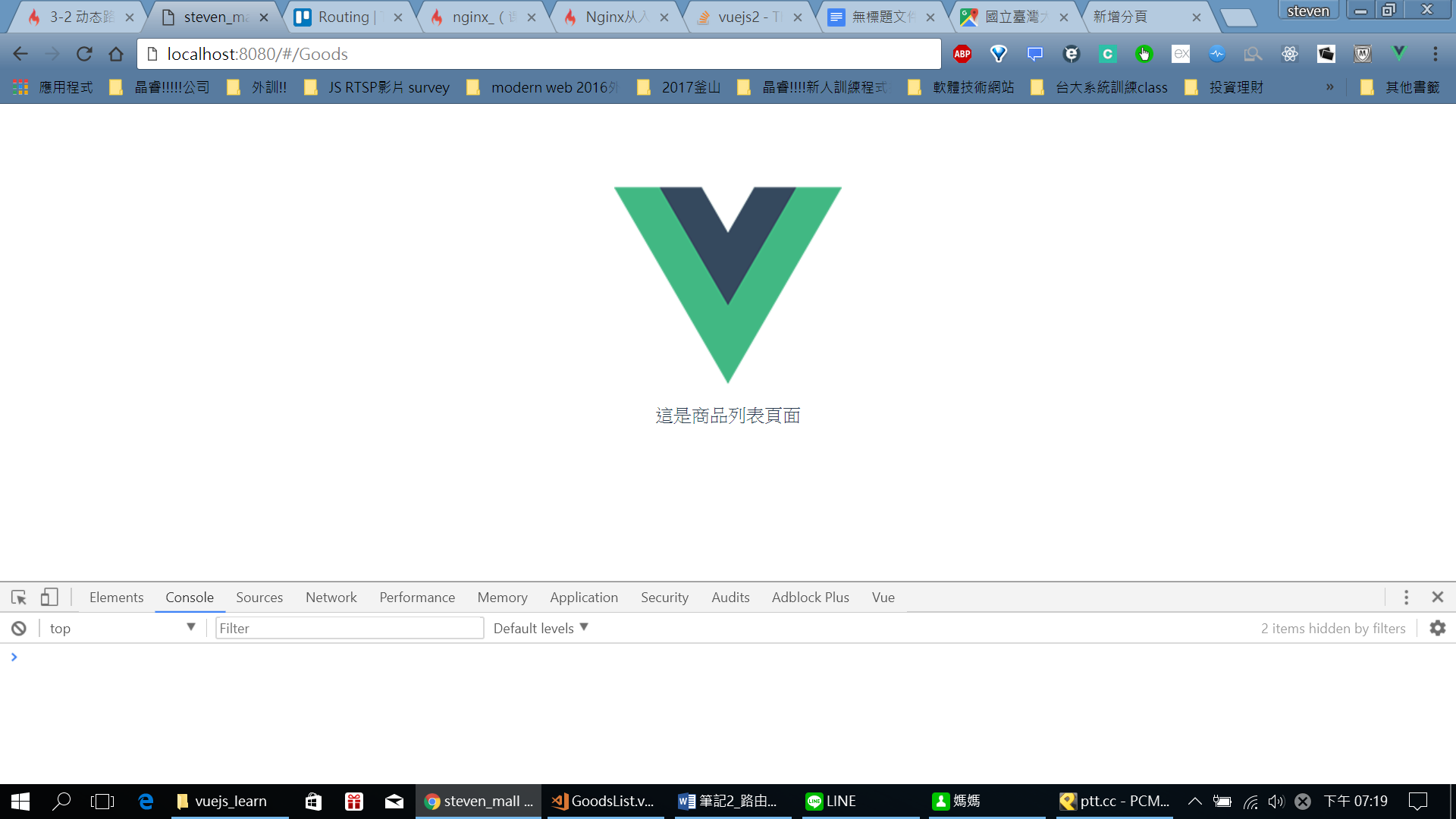
通常components都是放一些會一直復用的(像是footer header等)

這裡新增views來放商品列表





連上<http://localhost:8080/#/Goods>



接下來實現動態路由: (必須符合規則)

將route改成

path: '/goods/:goodsId',

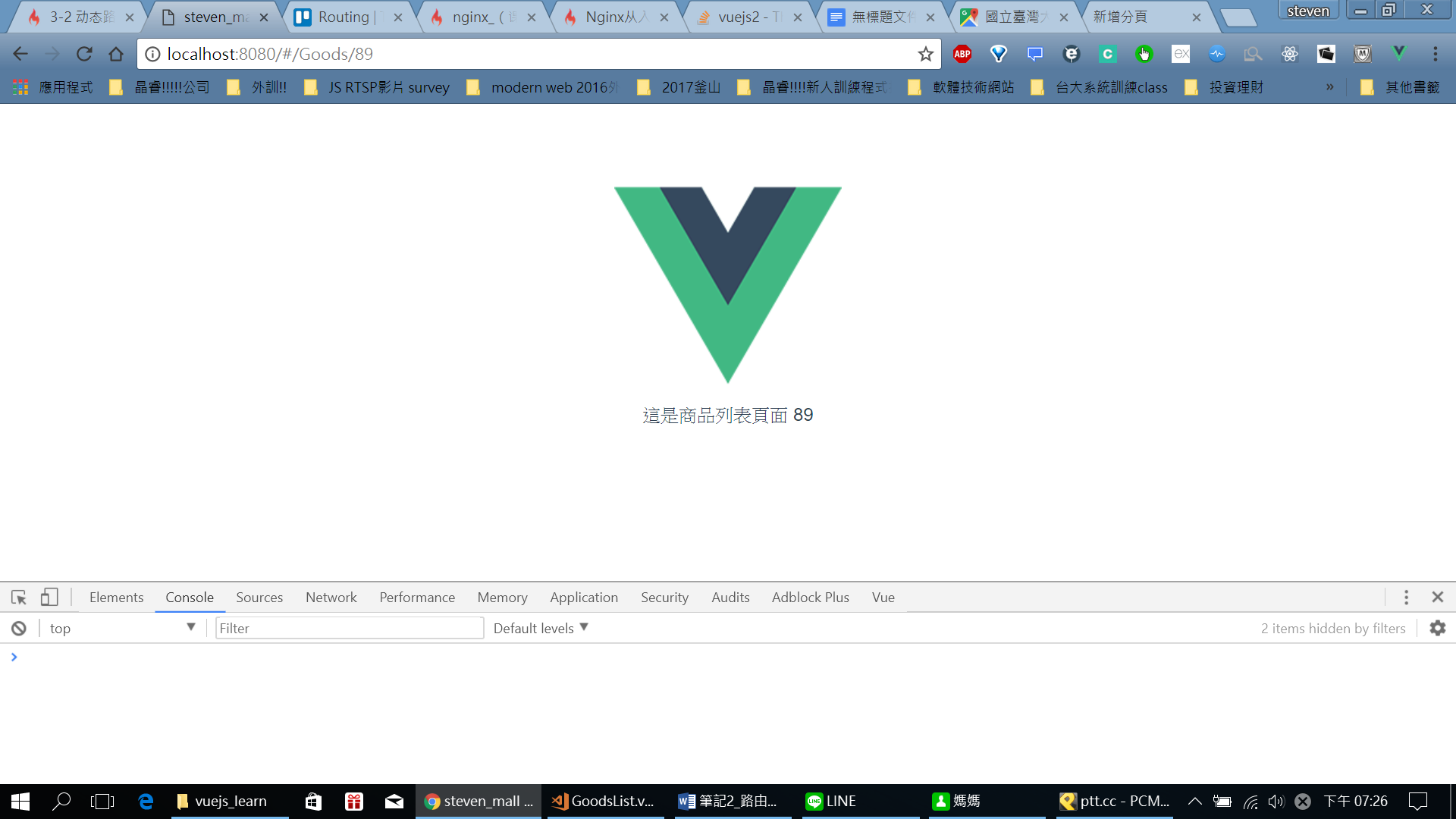
再將GoodsList新增:

<span>{{$route.params.goodsId}}</span>

此能抓到 剛route的goodsId

可再根據這個id查商品訊息 價格等等

此時UI必須在url上加上字才能連上:



若url指輸入<http://localhost:8080/#/Goods>

就沒有那行字了!

若再將路由改為:

path: '/goods/:goodsId/user/:name', //動態路由

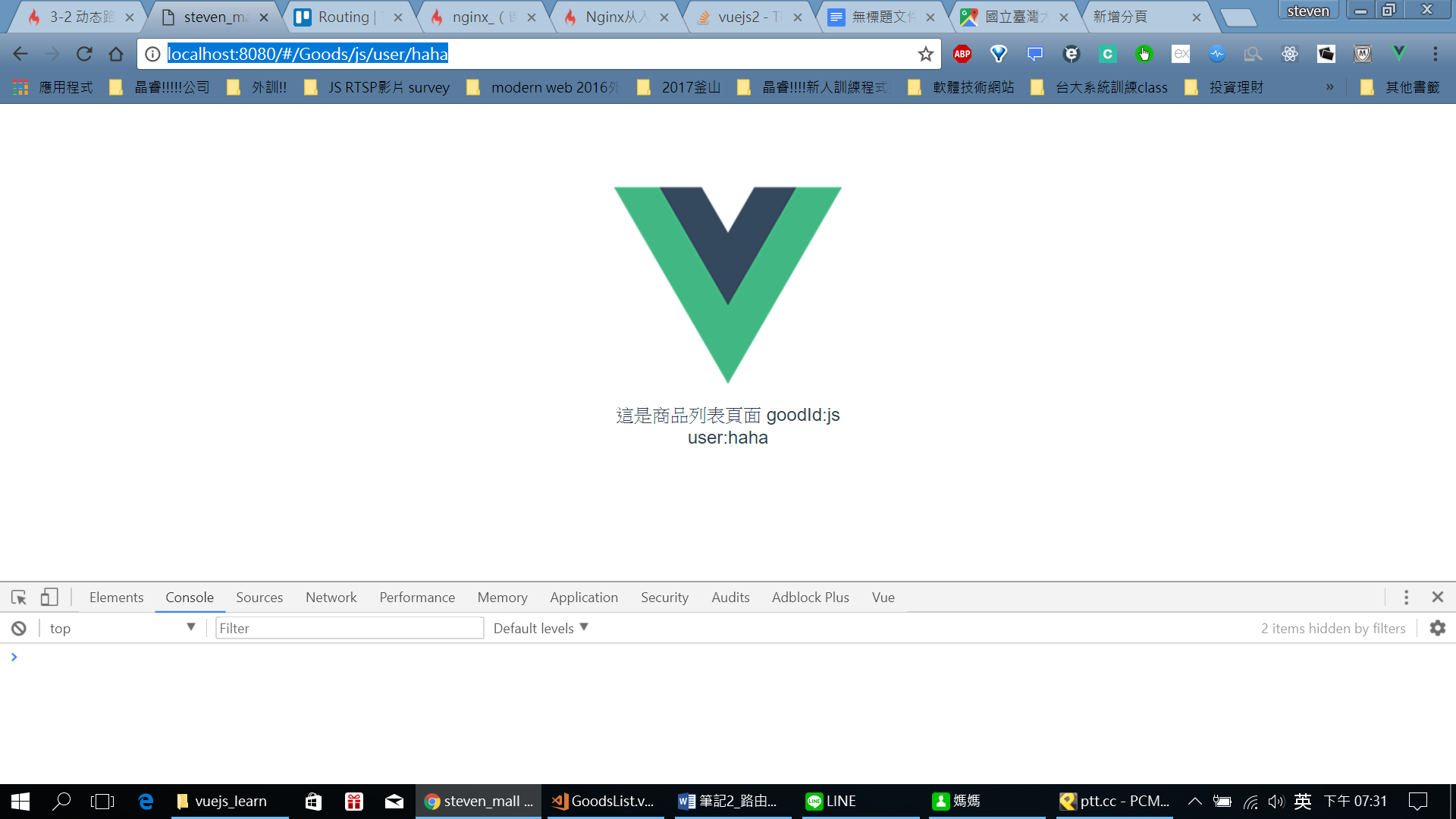
components寫上:

<div>這是商品列表頁面

goodId:<span>{{$route.params.goodsId}}</span><br>

user:<span>{{$route.params.name}}</span>

</div>



但會看到url前面有一個”#” >> hash

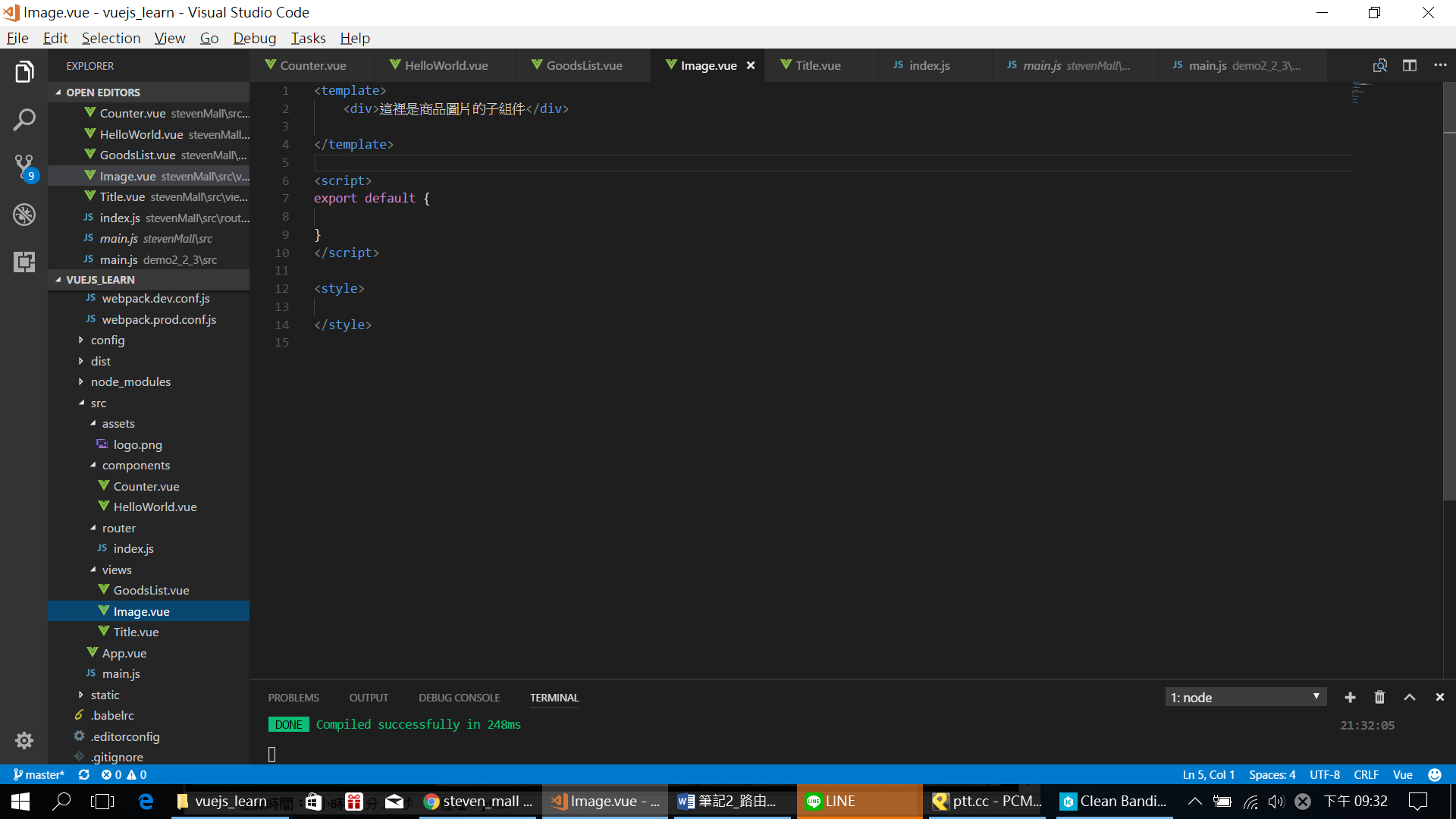
如果不要#就在route那邊加上mode:”history”

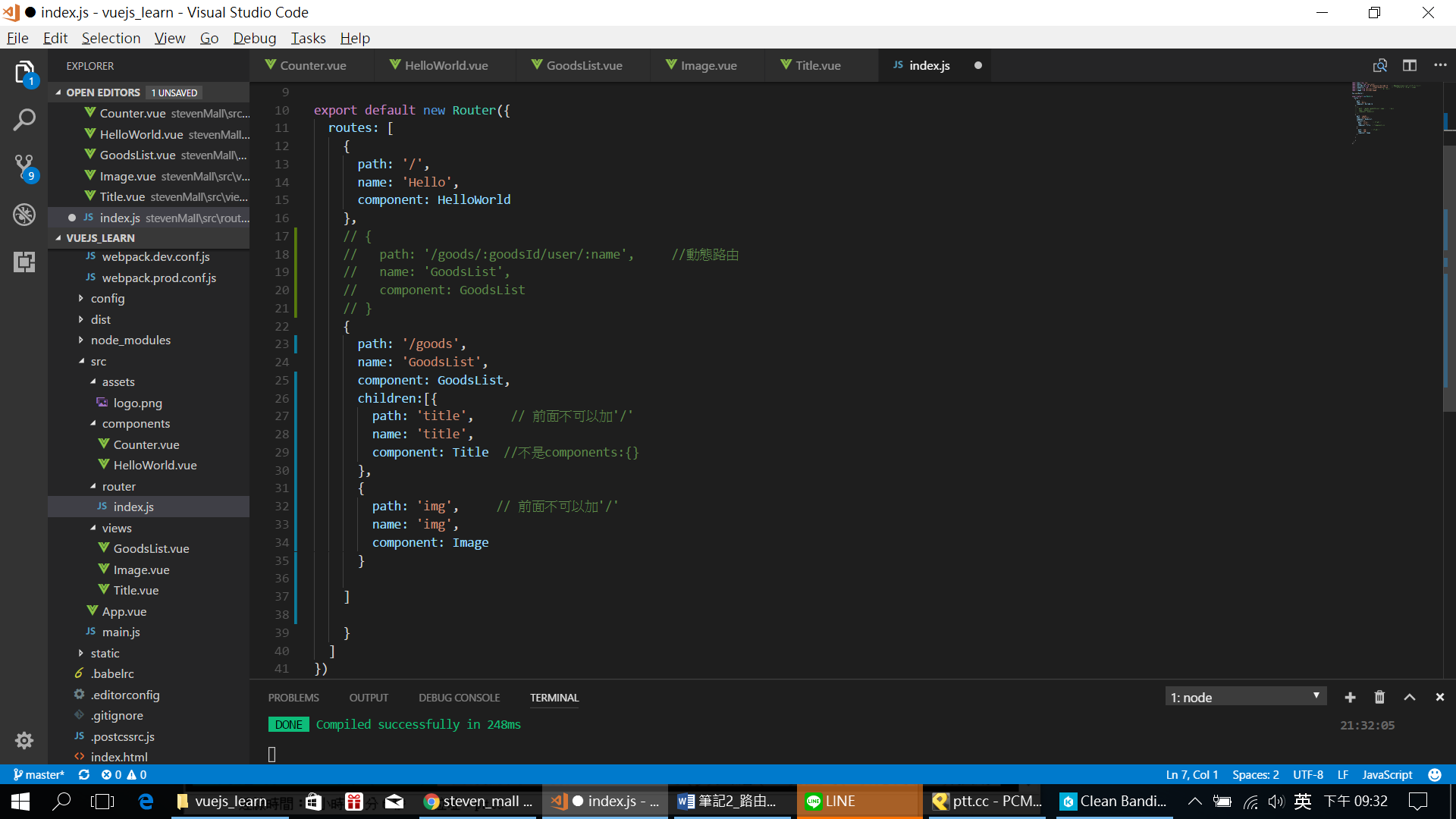
<可參考: <https://segmentfault.com/q/1010000010024213>>

* <3-3 嵌套路由[**Nested Routes**](https://router.vuejs.org/en/essentials/nested-routes.html)> 類似: 如果想要在同一個頁面，不同菜單的切換!!!!!

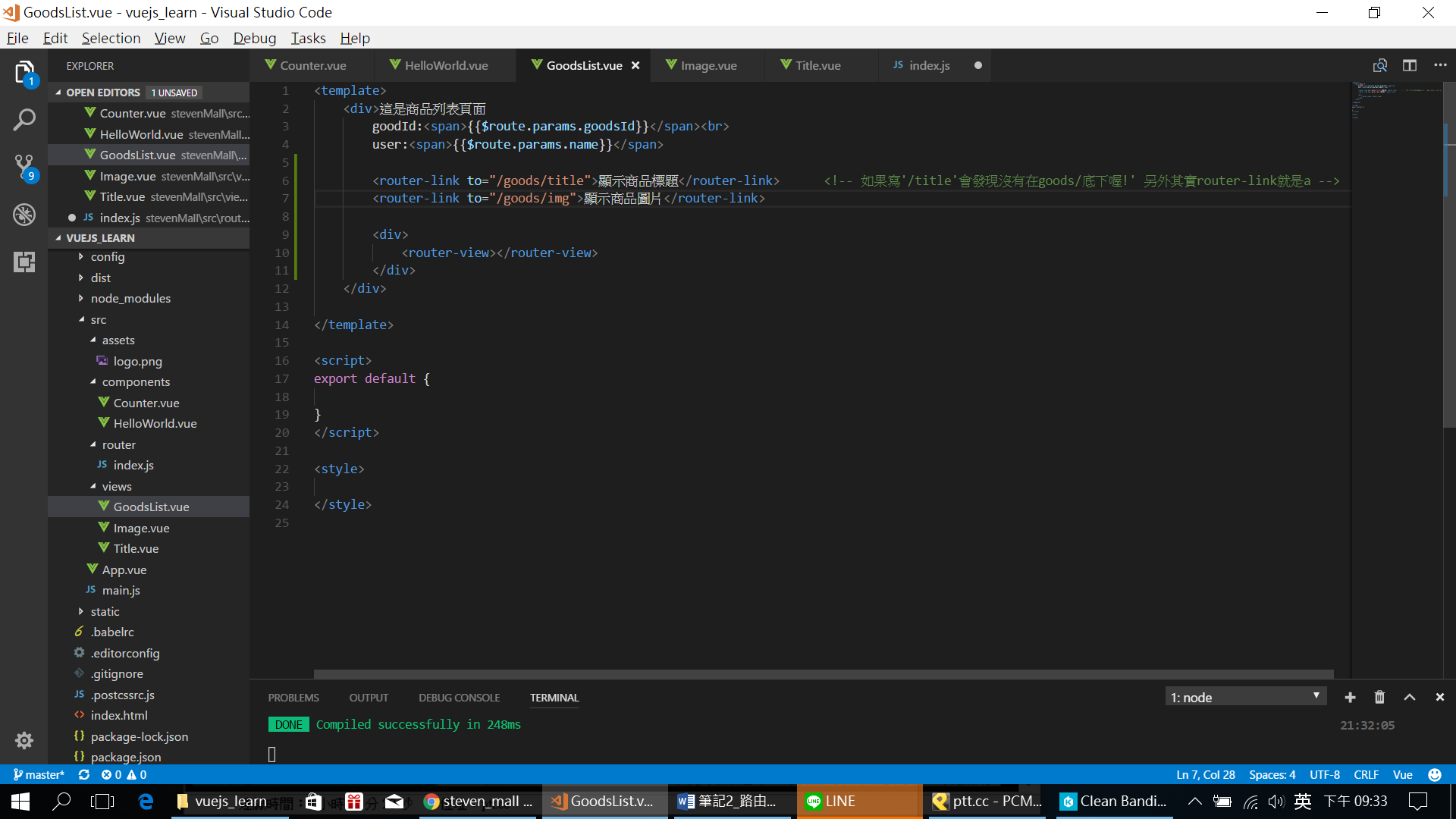
1. 接下來想要示範新增2個連結分別是商品Title/ image

像image:

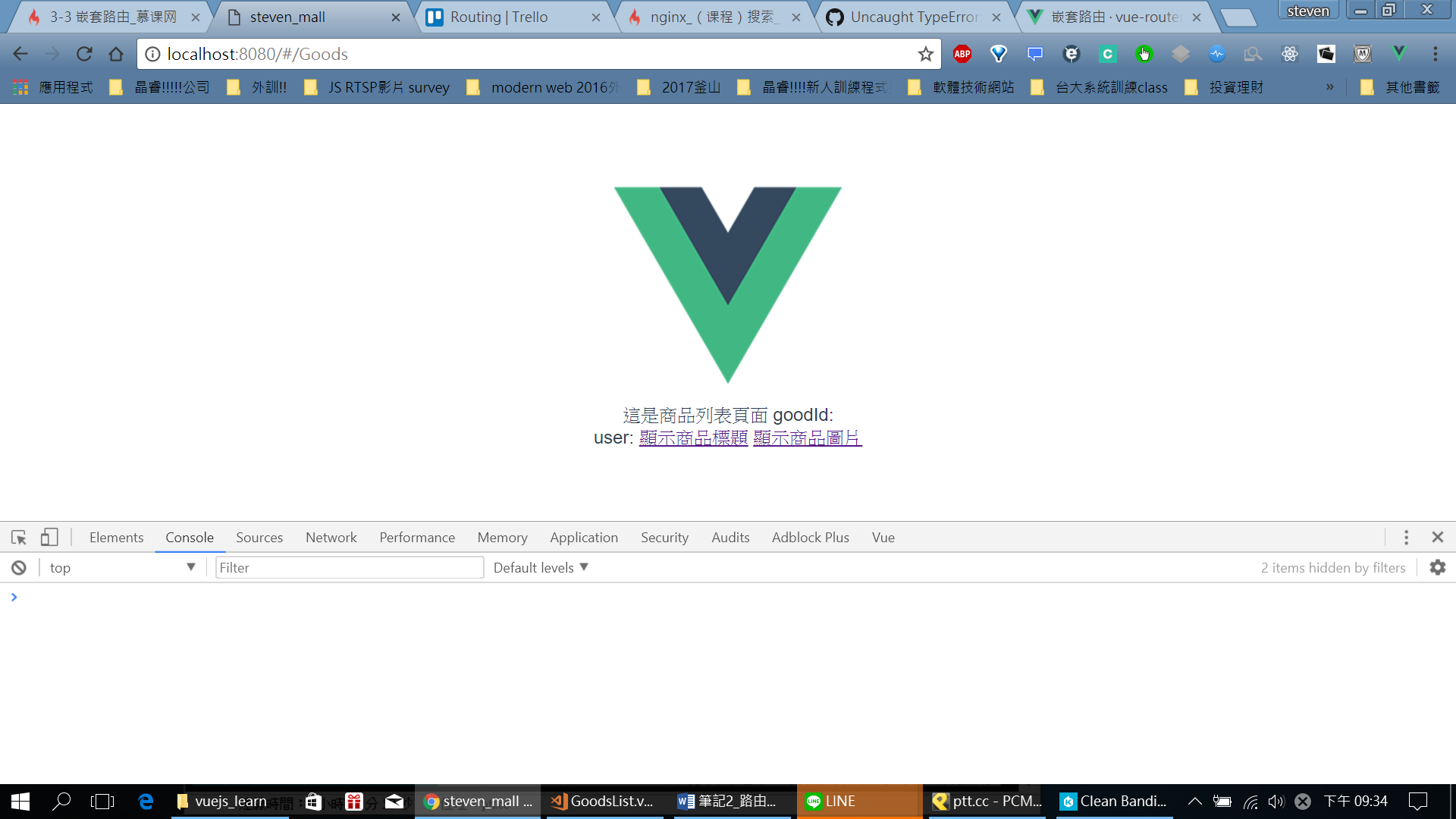


1. 路由改為:
2. 
3. GoodsLink新增router-link:

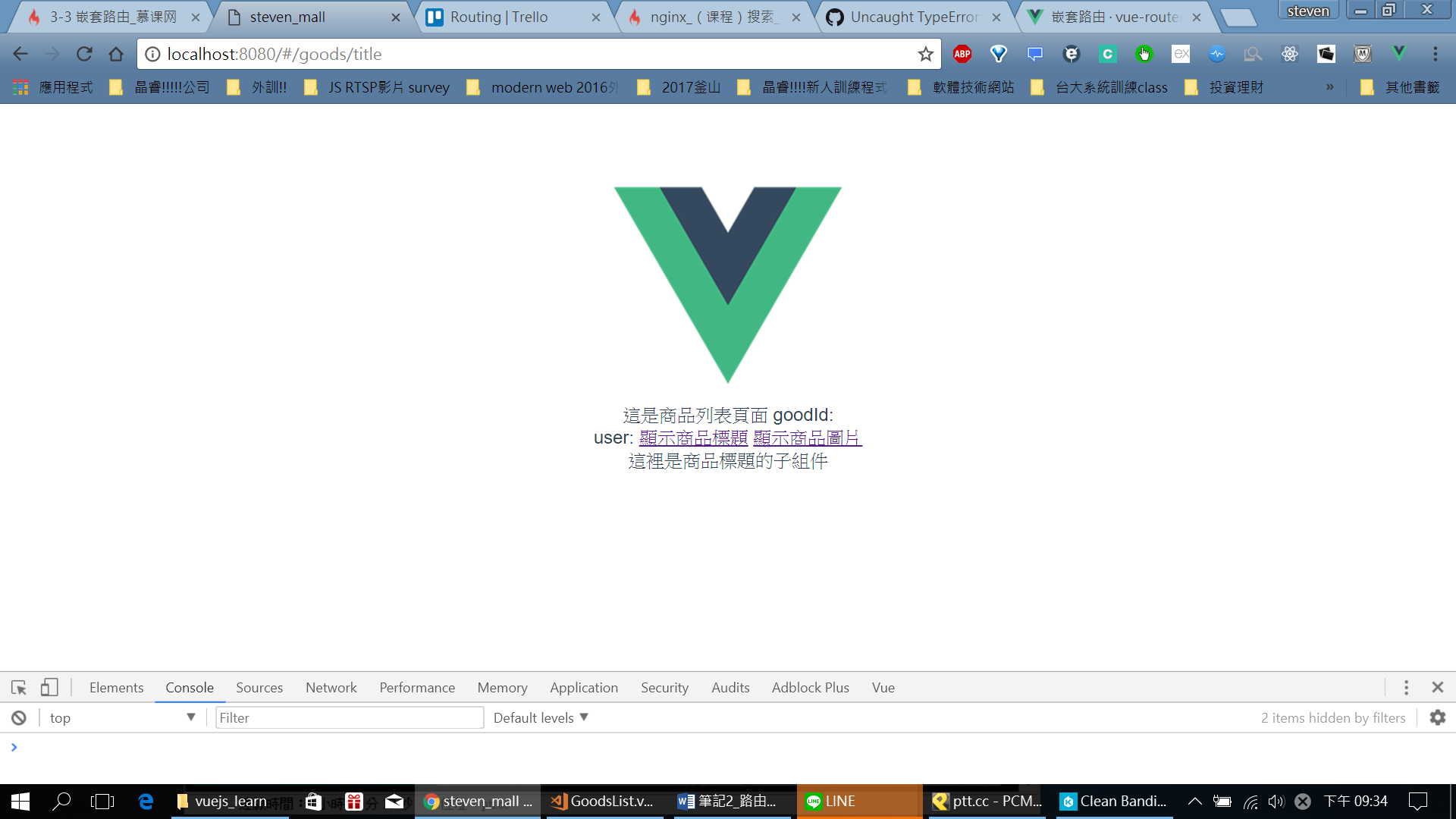
而router-view即為渲染router-link的顯示部分

1. 

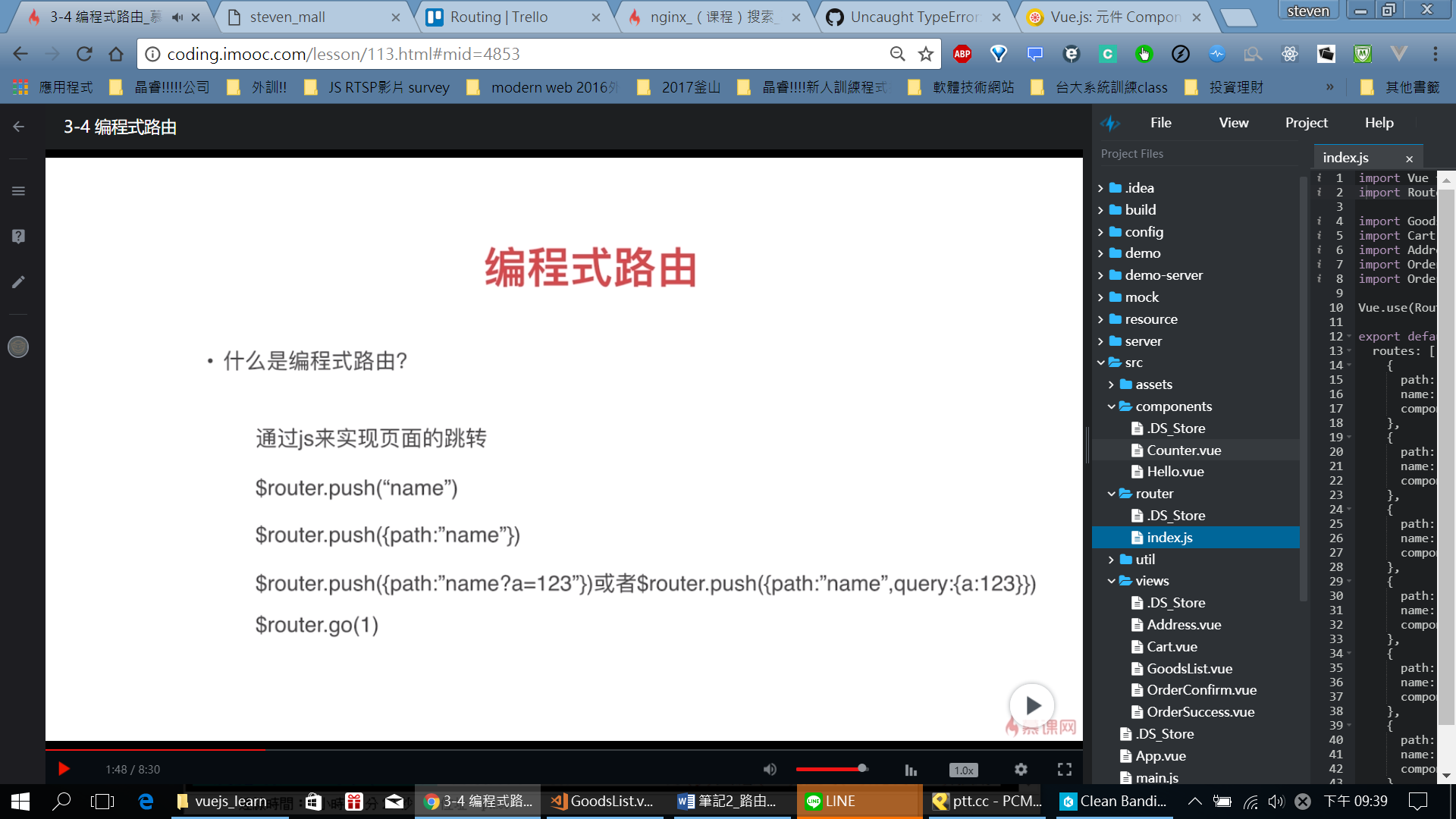
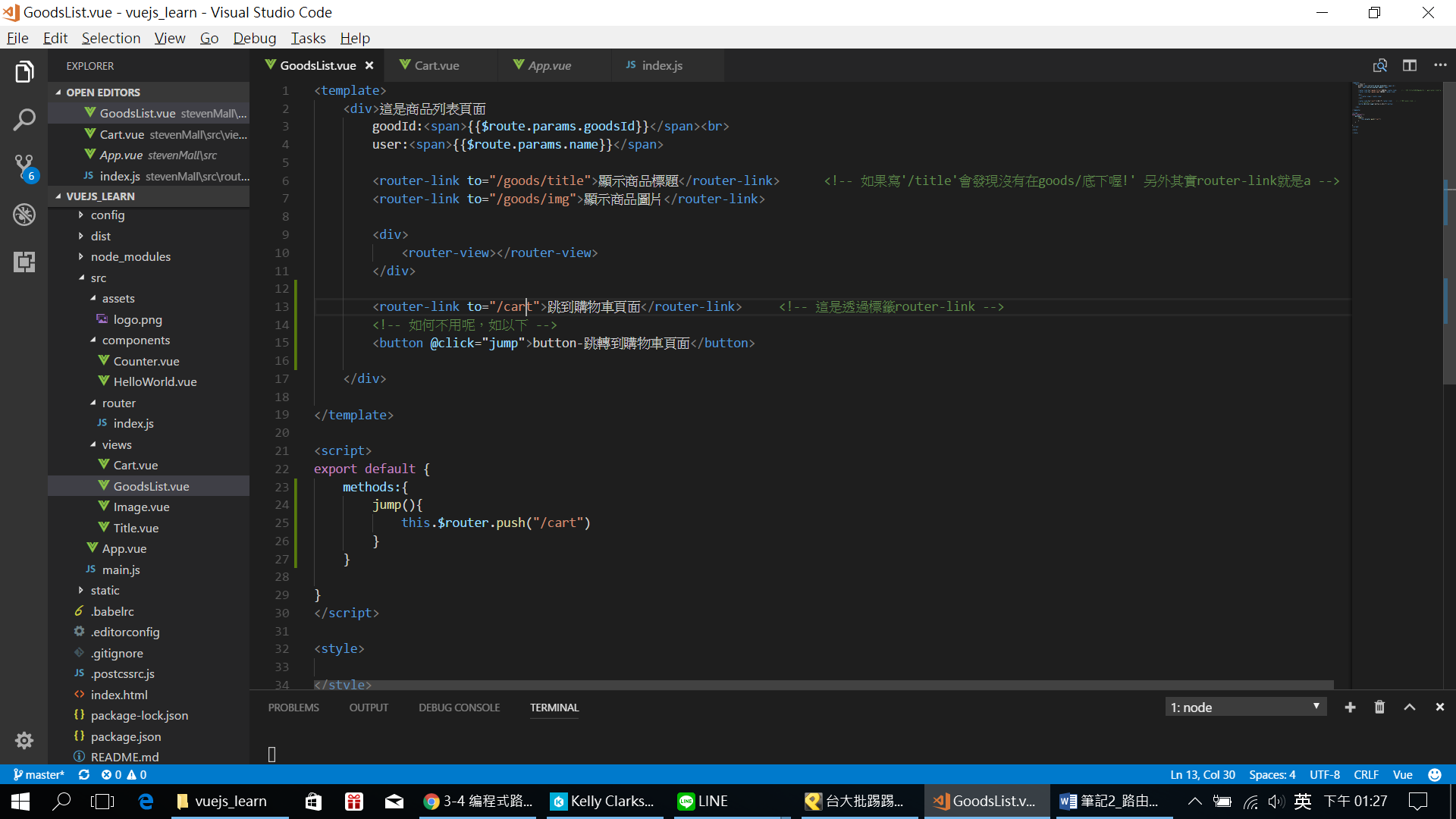
Router-link其實就是<a> to=’XXXX不可只寫children的path’

此時UI本來是

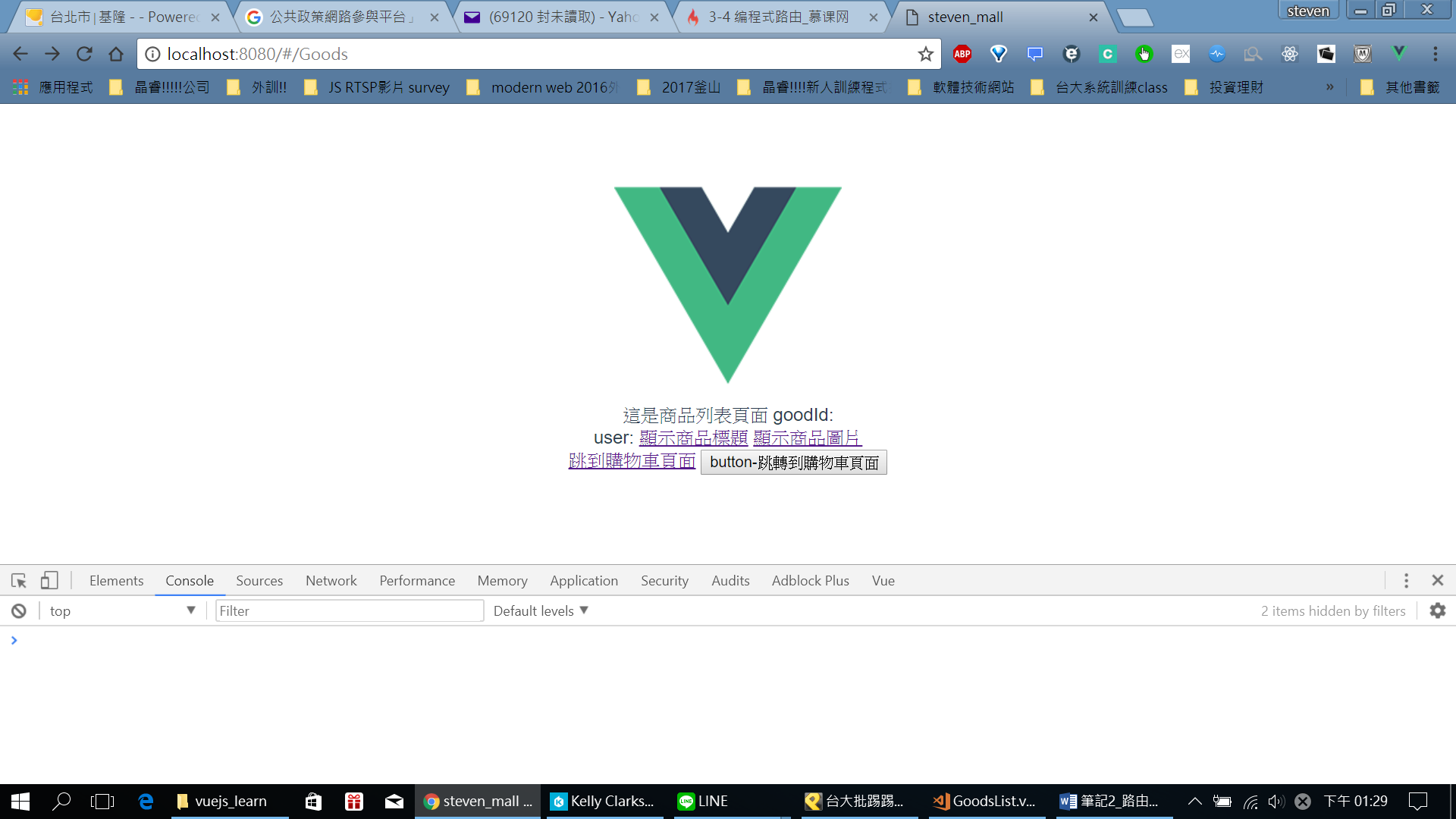
點選其中一個就會類似:

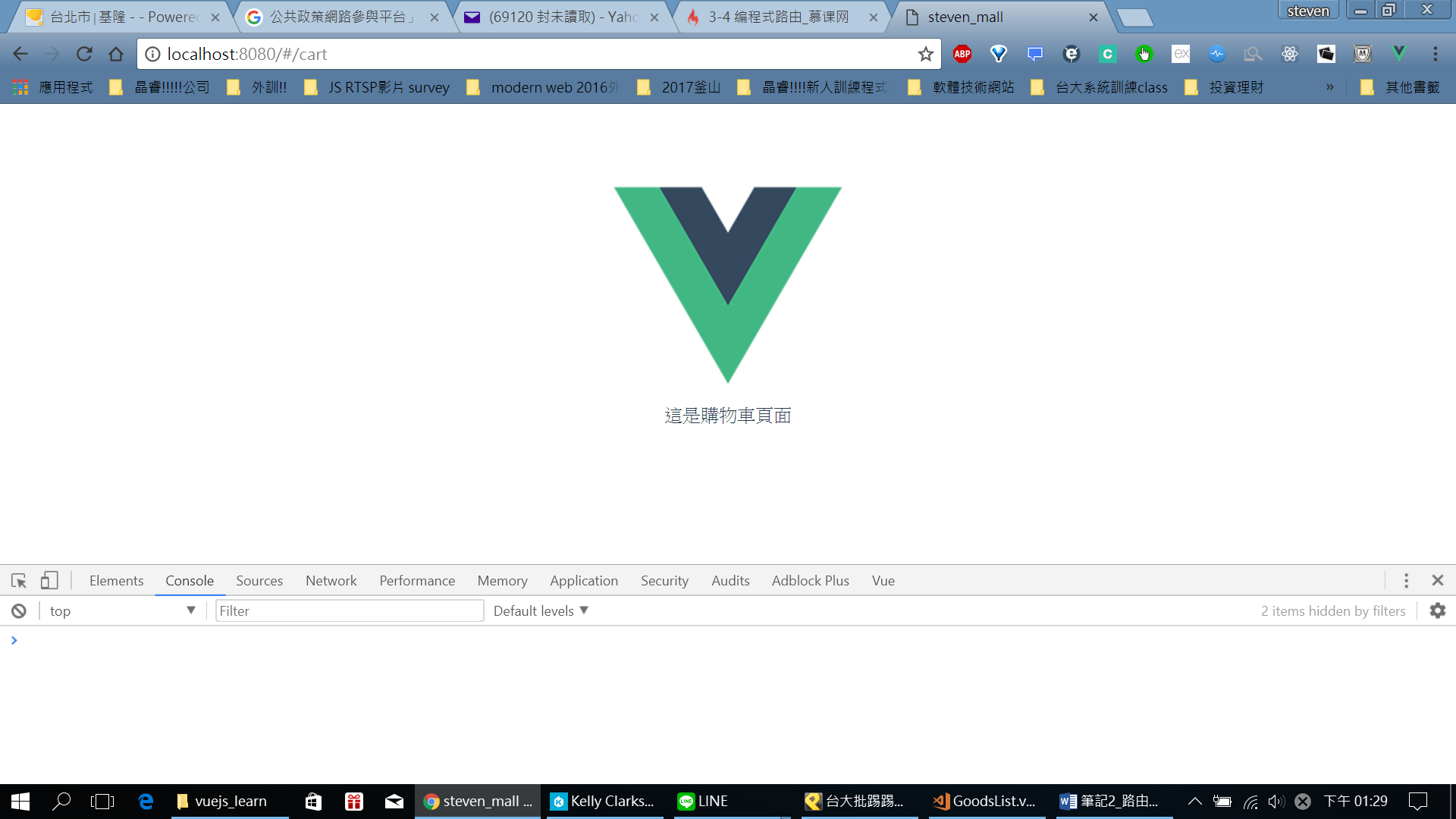


<編程式路由>

1. 即通過寫程式 (js):
   1. 
2. 例如在GoodsList:
   1. 

即透過@click,並在script的method寫上this.$router.push(“/cart”) 對應到router的index.js中的路由





也可以用:

jump(){

// this.$router.push("/cart") // 此為法一

this.$router.push({path:'/cart'}) // 此為法-4

}

this.$router.push({path:'/cart?goodsId=123'}) // 此為可以拿來傳參數

// this.router是在定義多路由~ 如果傳參數到某個路由，該路由要取某個參數則是使用$route.query.XXXX!

如何接收參數呢?:

很簡單的:

先到購物車頁面>

新增:

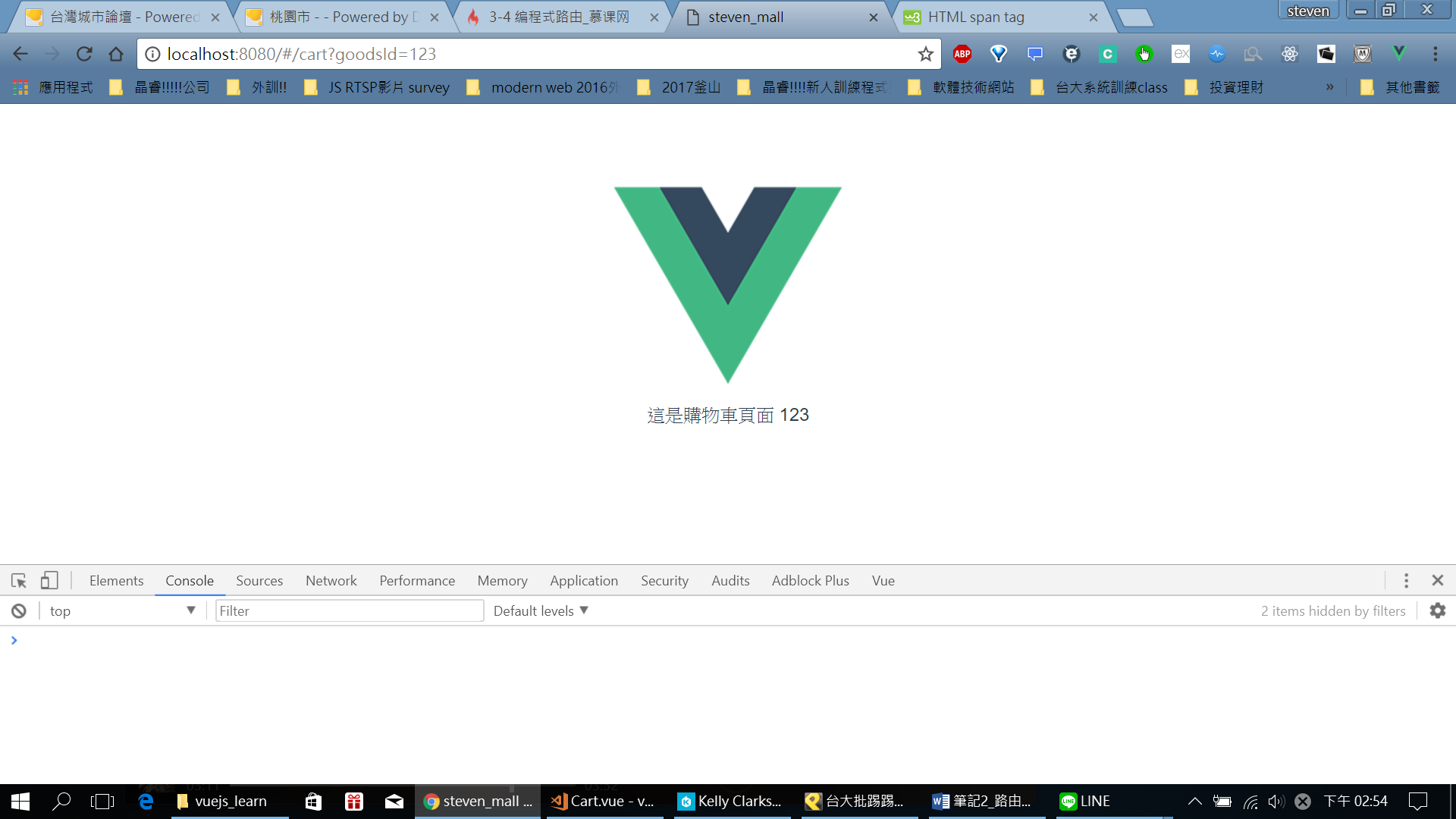
<div>

這是購物車頁面

<span>{{$route.query.goodsId}}</span>

</div>

UI就可以顯示剛$router.push({XXXXX?goodsId=XXXXXX})的XXXXX了



注意!!!

<div>

這是購物車頁面

<span>{{$route.query.goodsId}}</span>

<!-- 應該可以發現跟前面使用動態路由取參數的方式不太一樣: 前面是使用{$route.params.goodsId -->

<!-- {$route.query.goodsId是 透國頁面傳遞直接使用"問號"傳遞 -->

<!-- $route.params.goodsId 則是動態路由在router的index.js的path: '/goods/:goodsId/user/:name', 的寫法在傳遞 -->

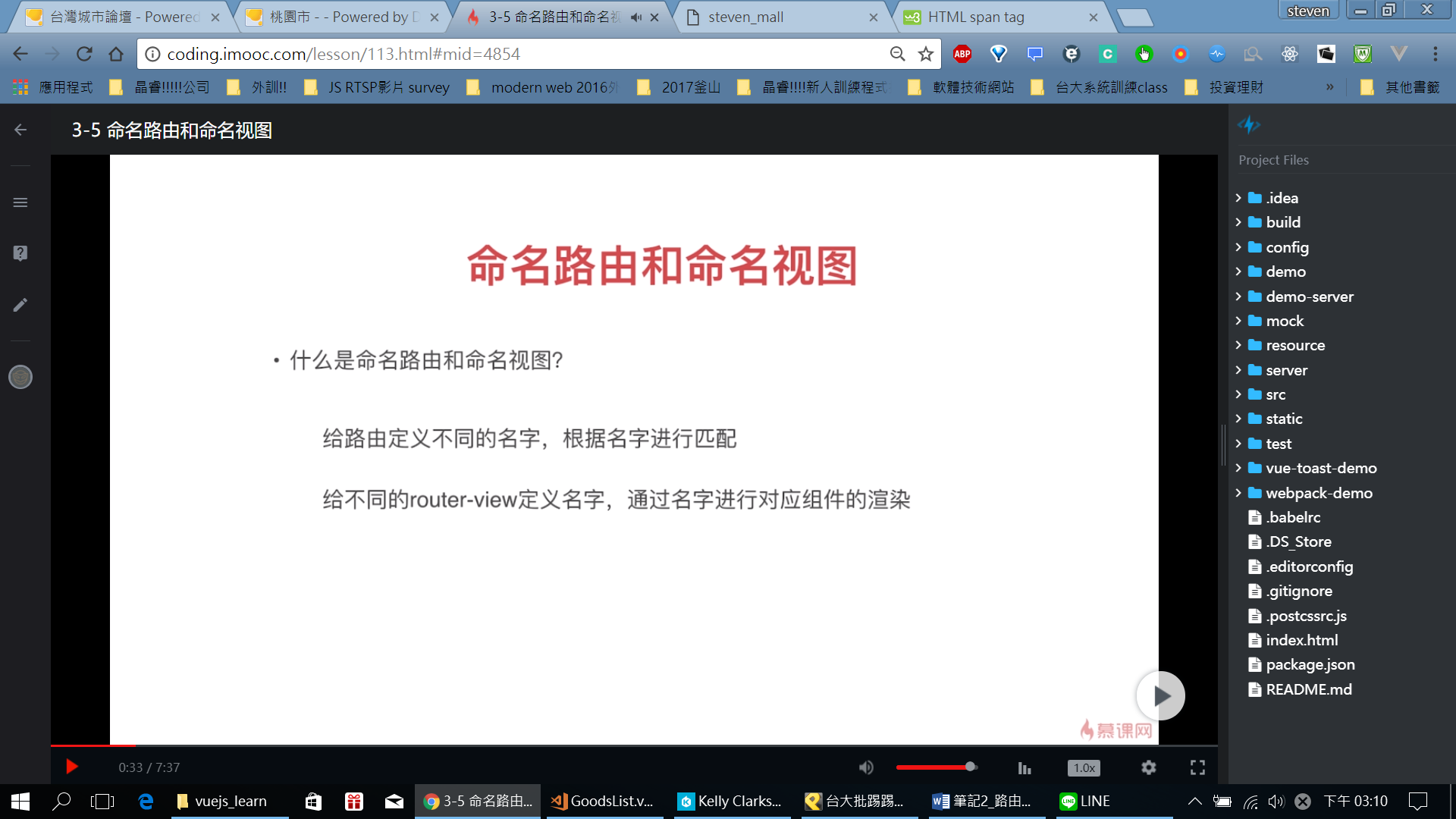
</div>

另外:

// 還可以透過.go去實現history往前頁往後頁的概念:

this.$router.go(-2); //和在console輸入history.go(-2)概念同

<命名路由和命名視圖>

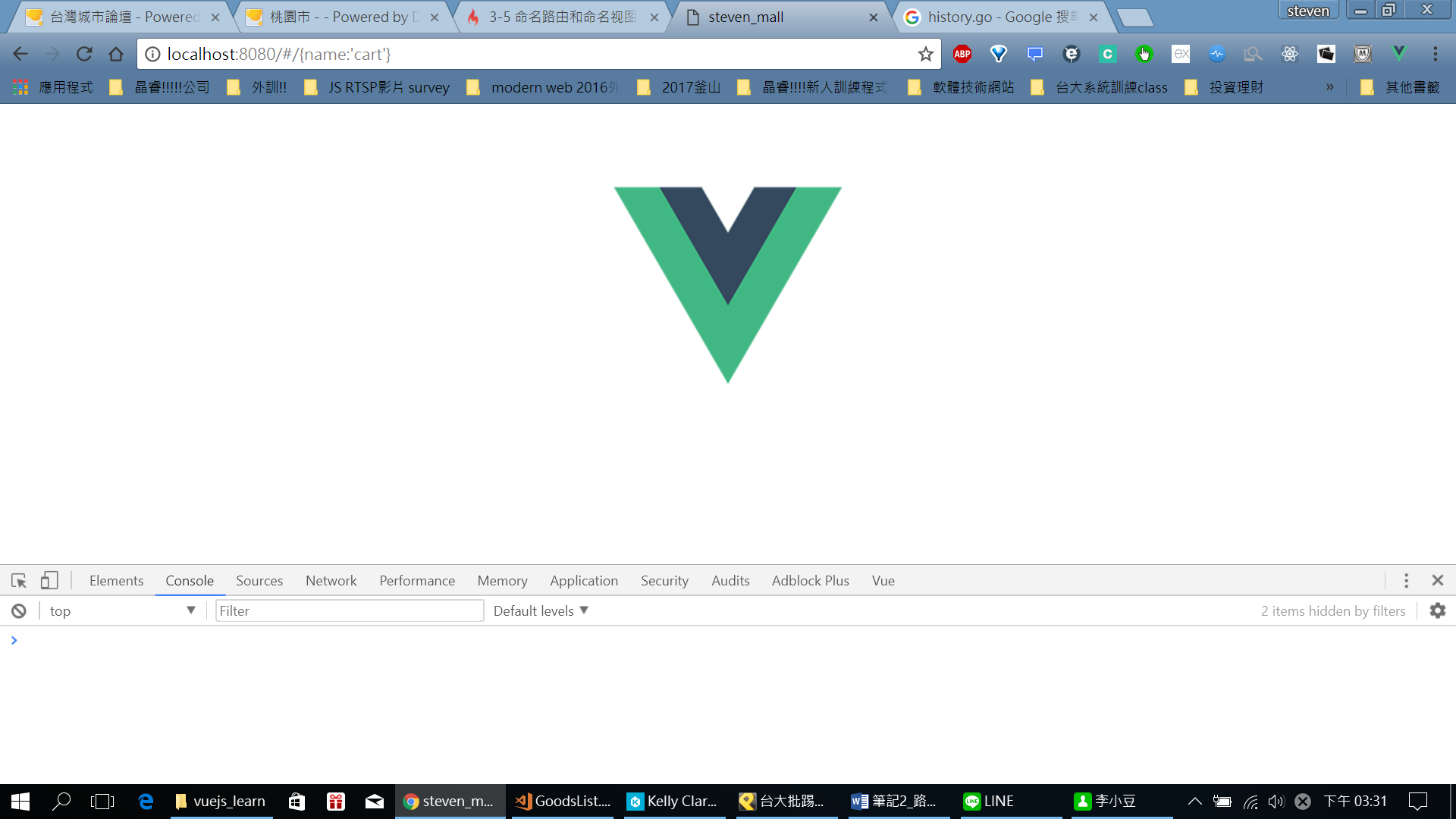


命名(name)路由亦即用我們 在router所定義的每個route的name來實現:

例如在GoodsList寫上:

<router-link to="{name:'cart'}">跳轉到購物車頁面 by name</router-link> <!-- 用name來指定路由,給一個含name的物件 -->

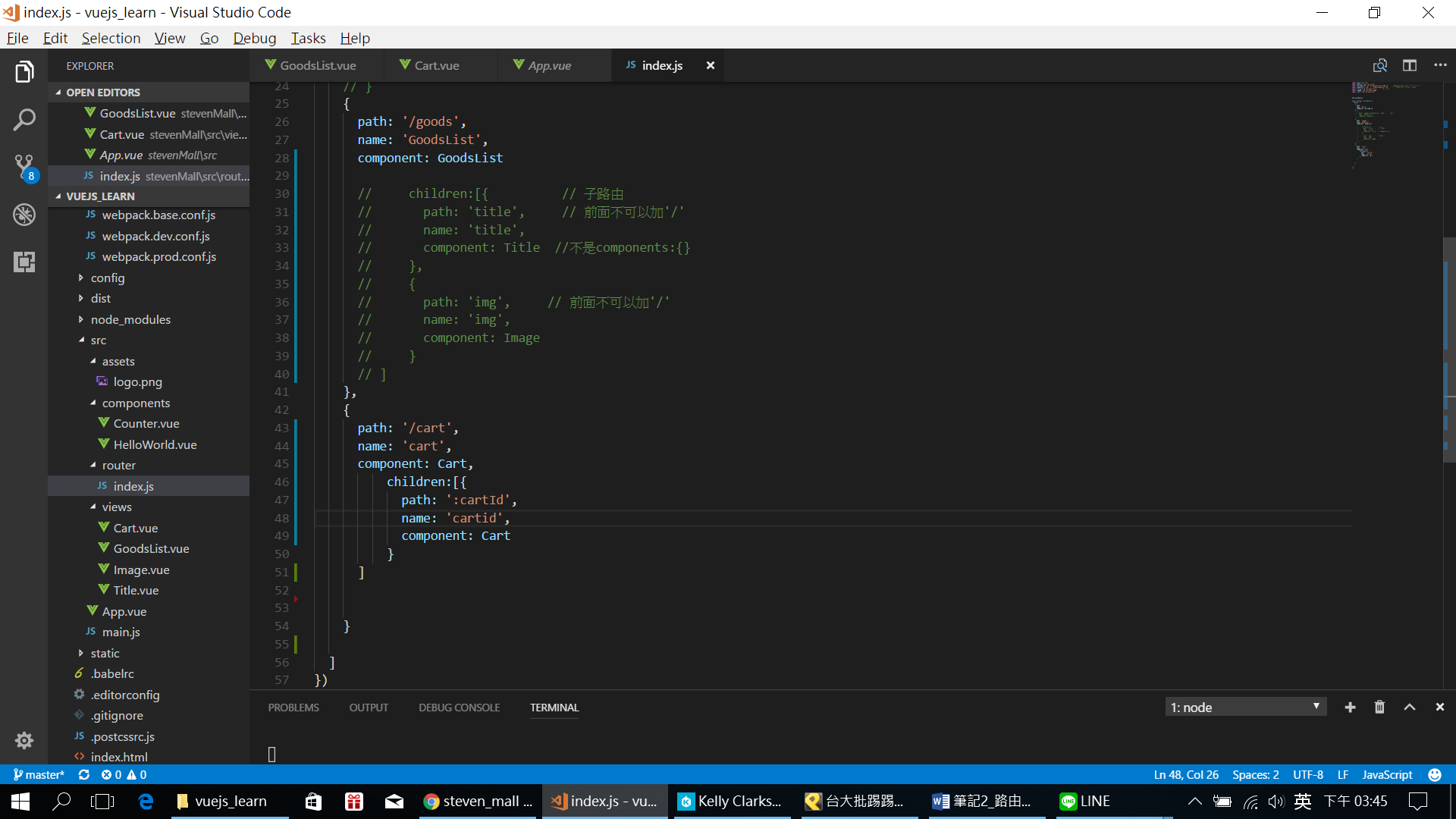
但是點擊後會發現一個坑:



即URL後面變成: <http://localhost:8080/#/{name:'cart'}>

所以應該要寫成>>>

My: 假如

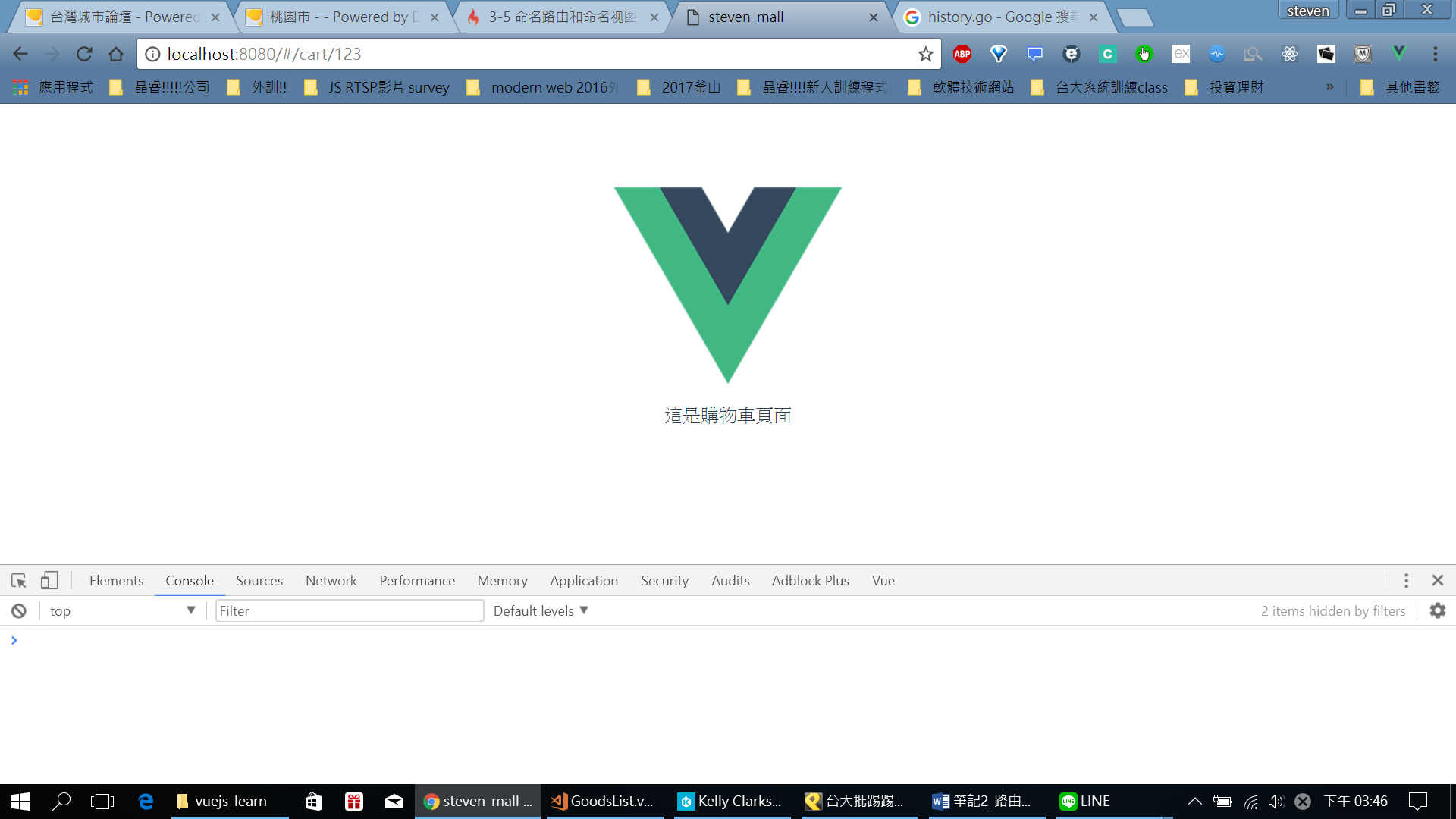


而購物車頁面為:

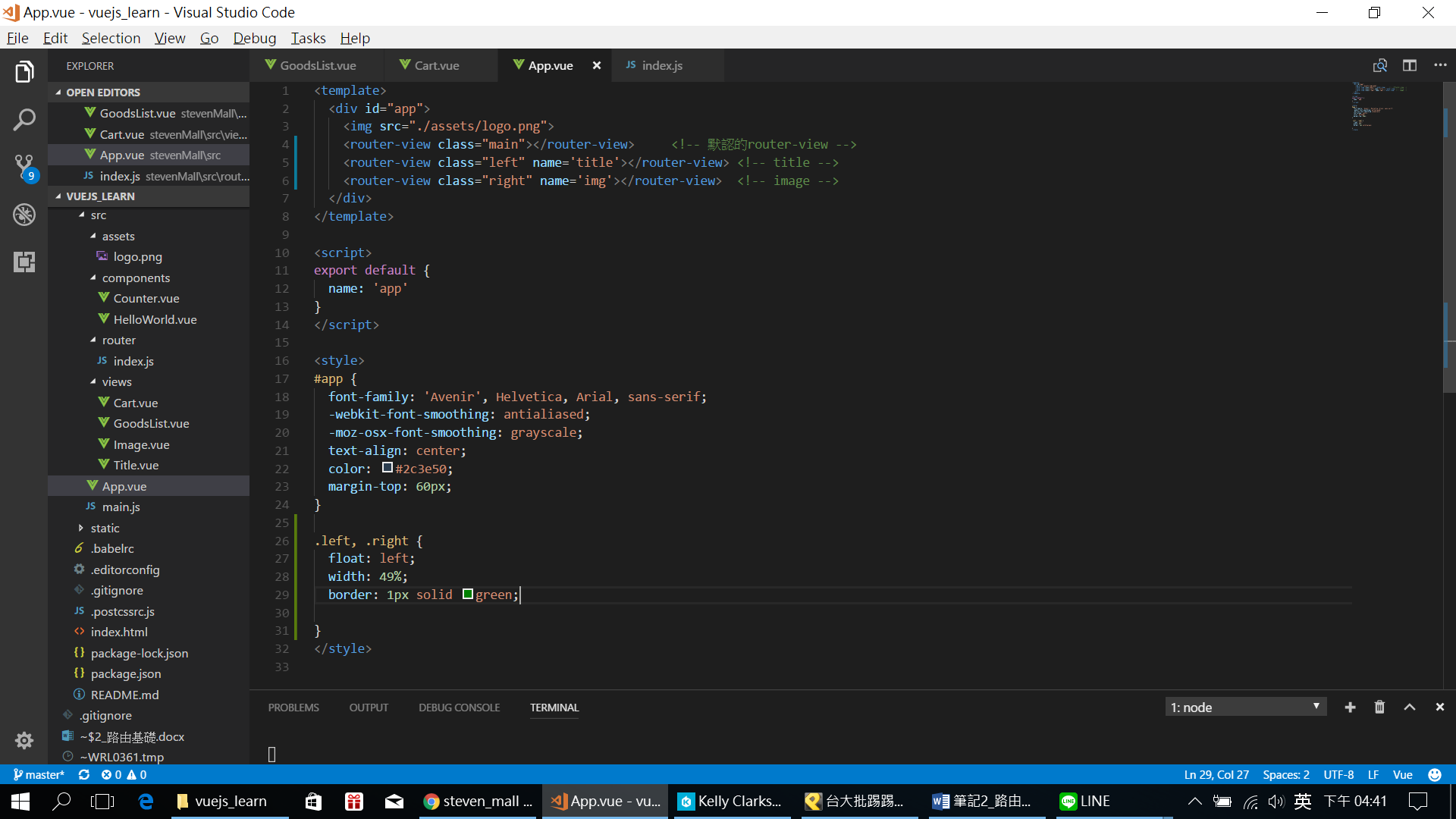
<!-- 要寫成>> v-bind: 如以下才會成功!-->

<router-link v-bind:to="{name:'cartid', params:{cartId:123}}">跳轉到購物車頁面 by name</router-link>

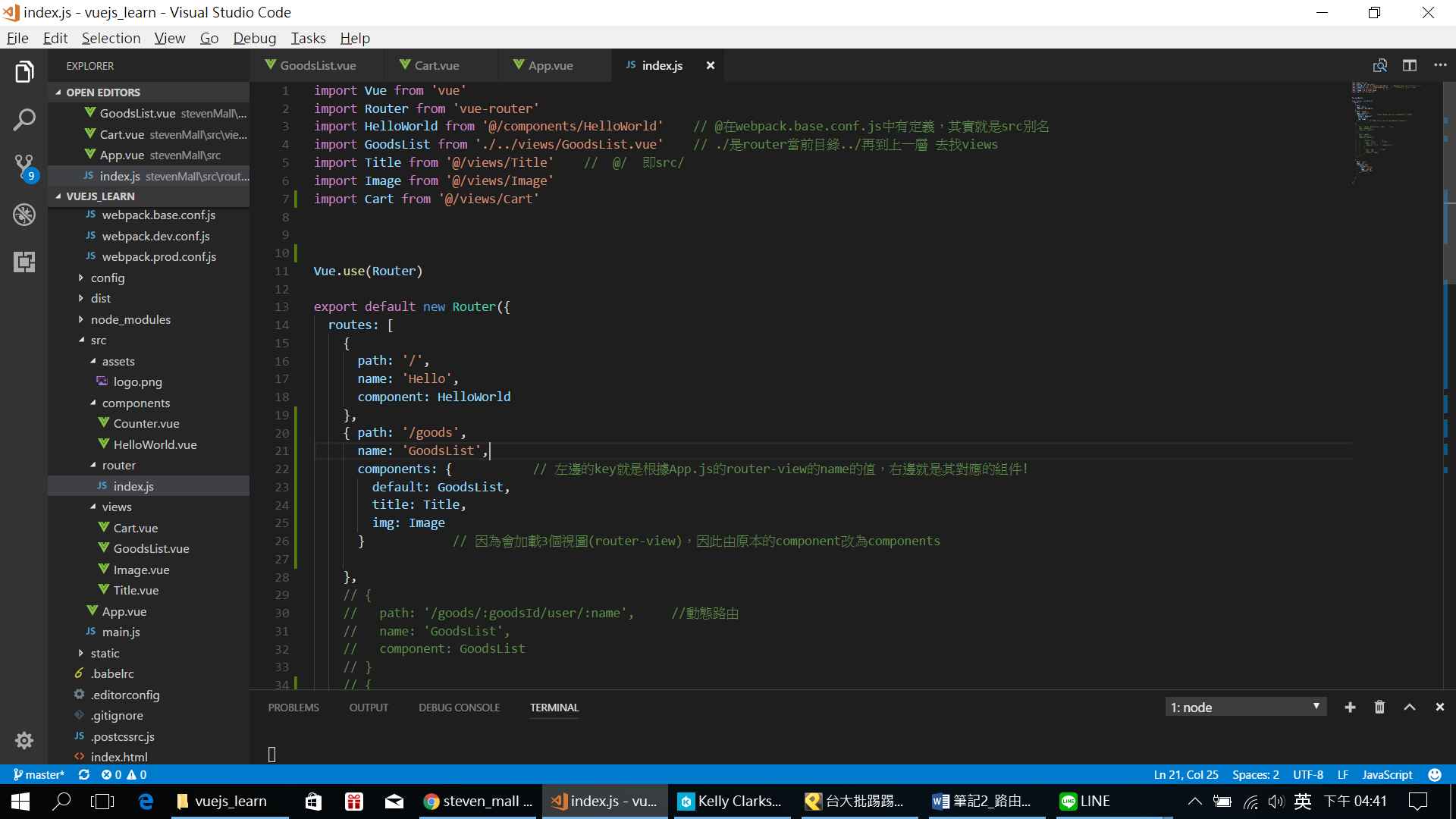
就成功可以跳轉並看到url正確看到ID了



接下來要講命名視圖:

* 1. 在App中的router-view加個name:
  2. 

Routers改為:



<因為router-view有3塊所以寫components而非component>

但其實這種命名視圖不常用

>>>

可以直接在商品列表頁面提供3種區塊就好呀

只是官方有提供想必也有它的可用之處吧!