<Vue Resources>

異步型插件

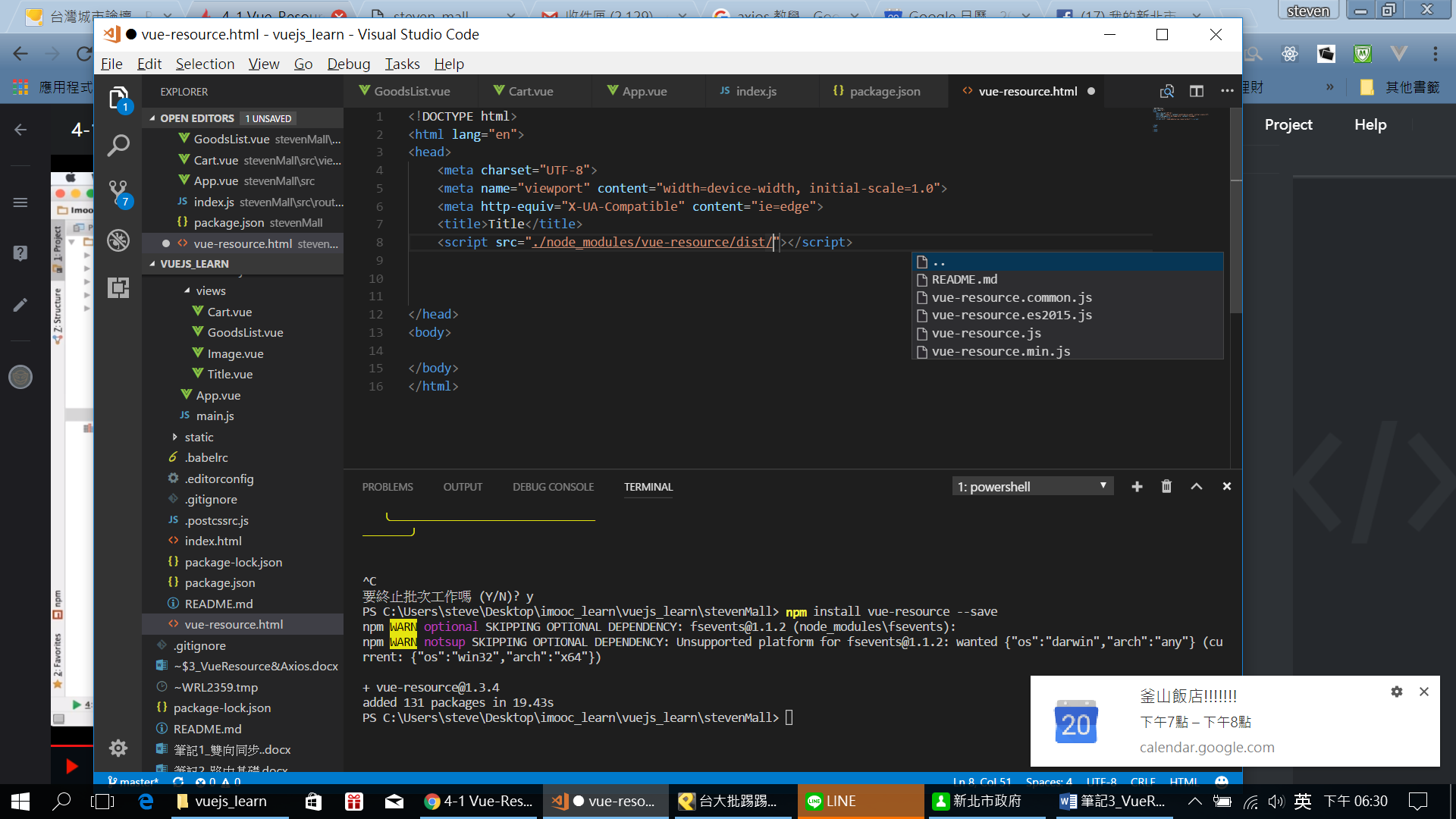




<可比較Jquery的AJAX和接下來的Vue-Resource!>

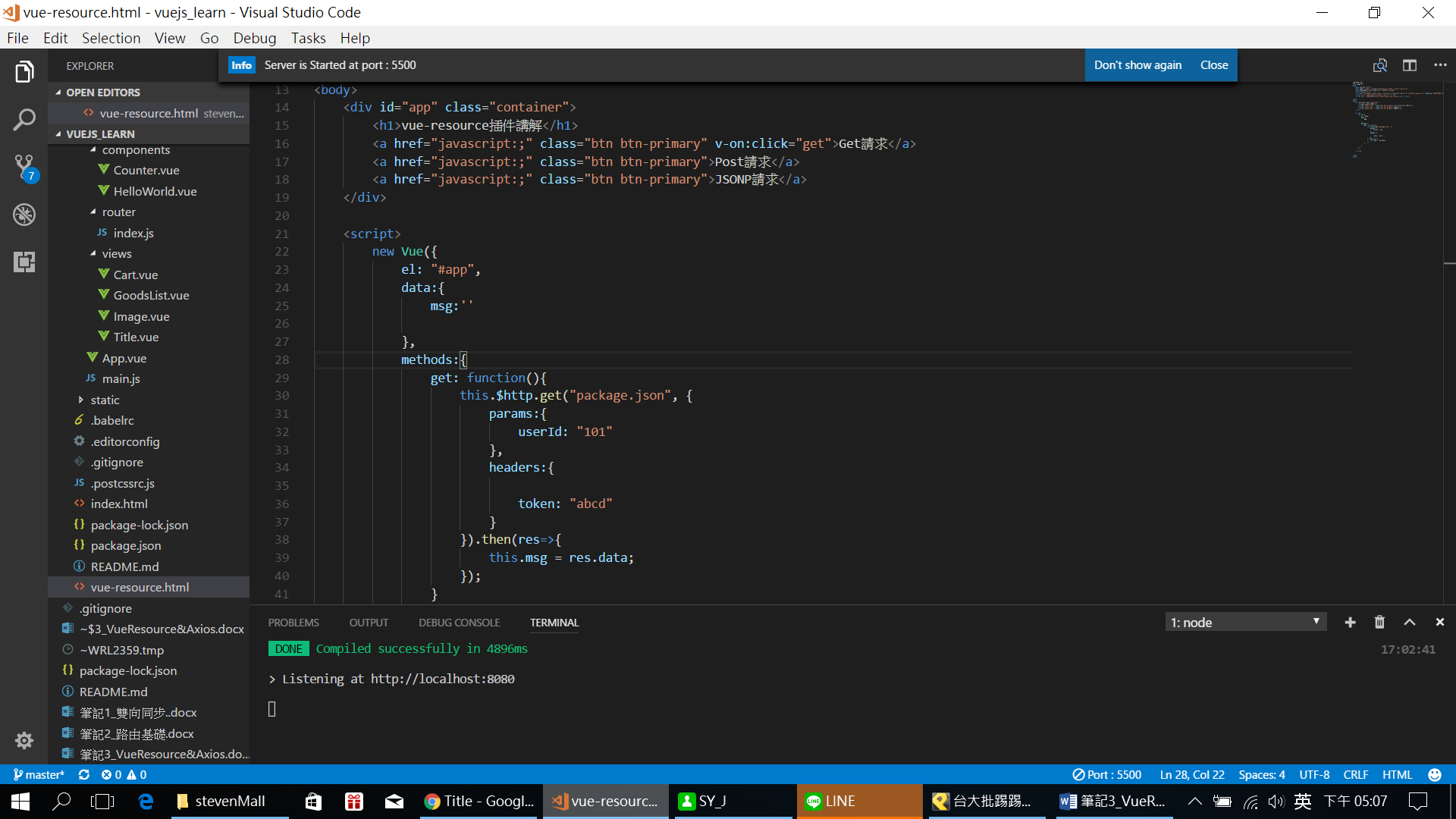
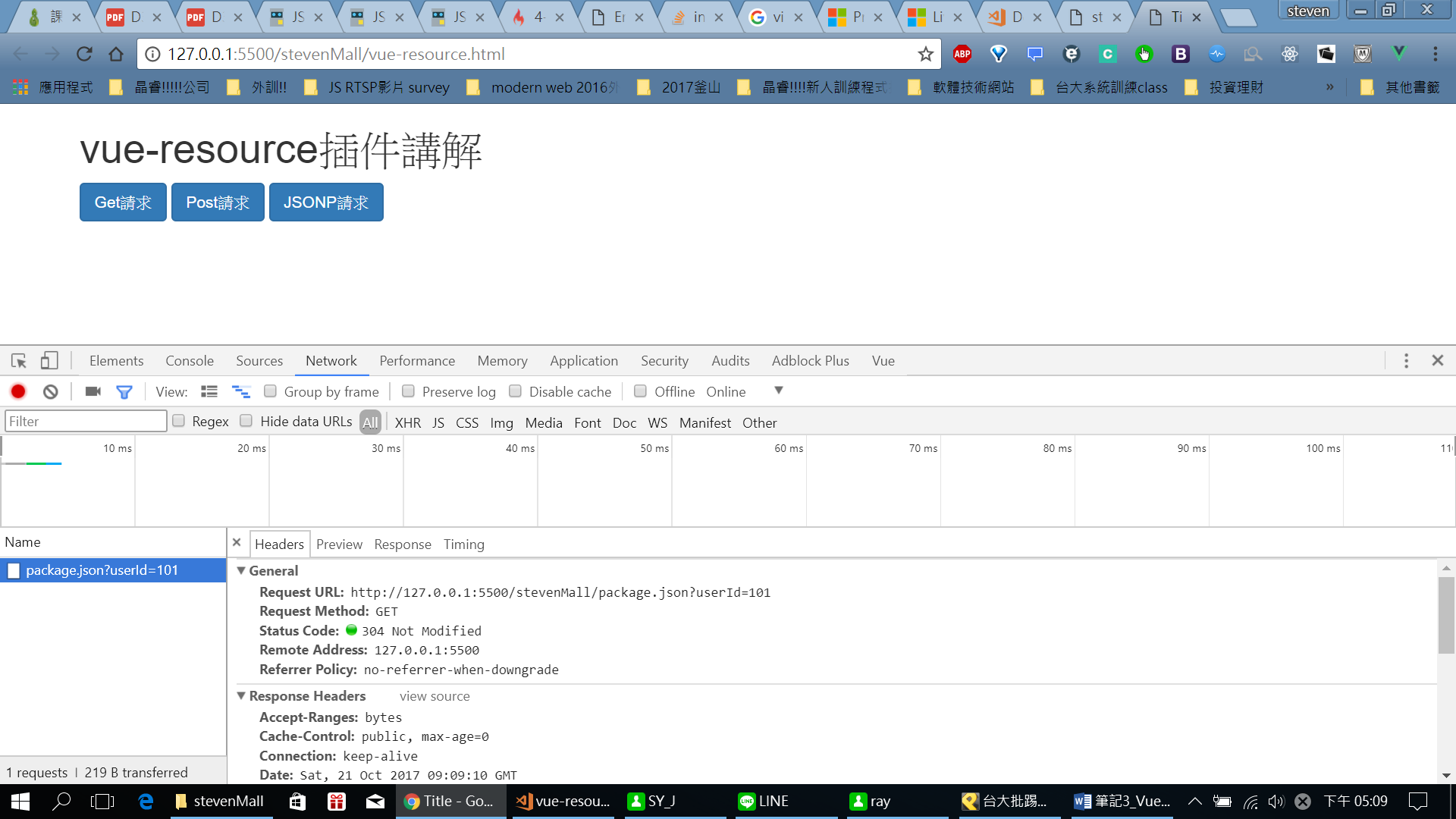
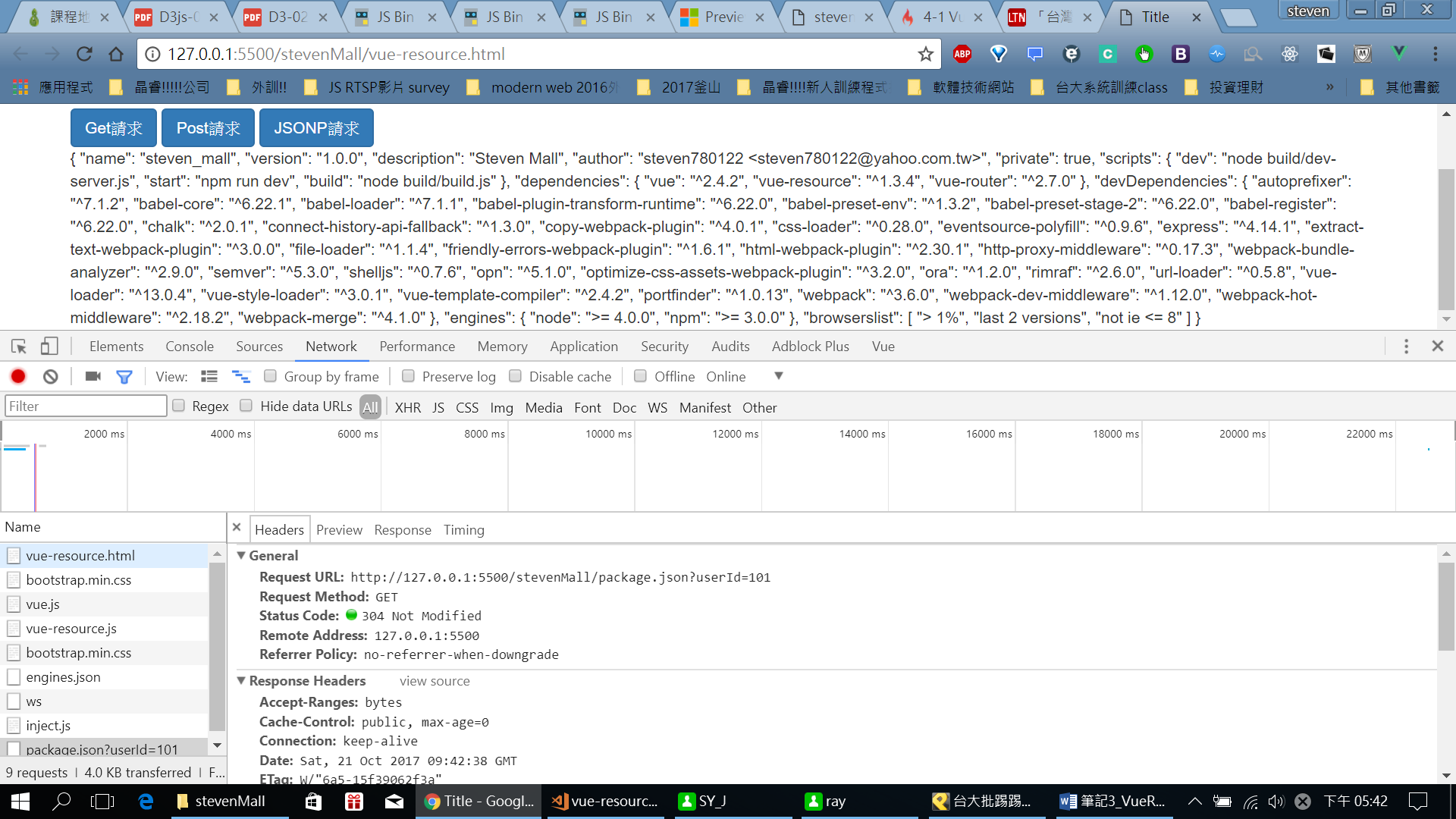
<https://ithelp.ithome.com.tw/articles/10188447>

1. 先npm 安裝vue resource到dependance (和devDependencies的差異就是開發時需要，還是打包後仍需要，例如babel(ES6解析) 在開發時打包視當然需要，但打包完後當就不需要了!)
   1. npm install vue-resource –save
2. 外層新增vue-resource.html:



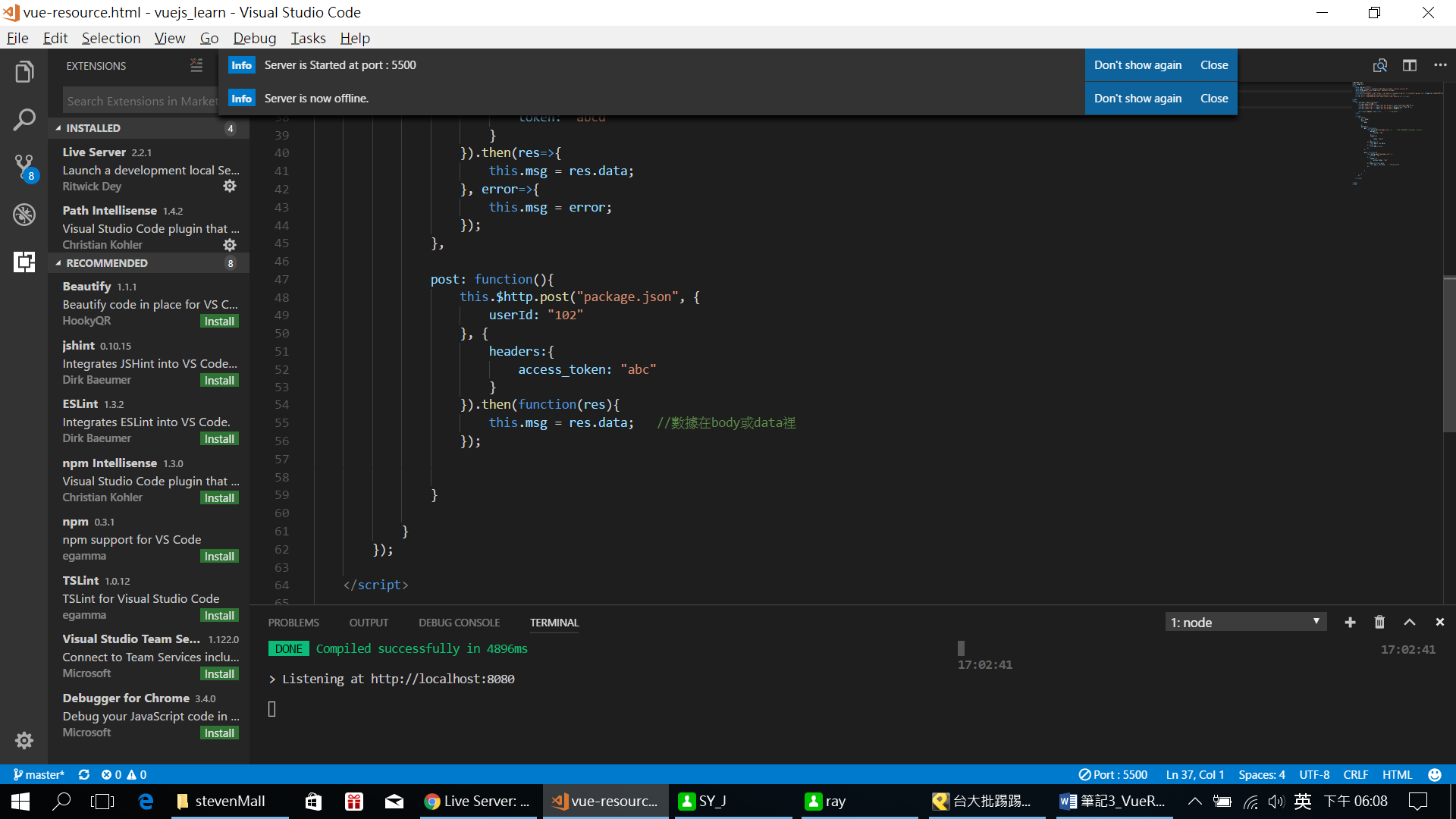
可以看到有4個檔案!

可以去查查區別 (來自不同的規範或是是否壓縮過)

1. 可參考bootcss.com
   1. 可到文擋去下載一個CDN地址
2. <a href="javascript:;" class="btn btn-primary">JSONP請求</a>
3. <a href="javascript:;" 用來防止刷新!
4. 在vue-resource輸入
   1. <GET>:
   2. 
   3. 可以在browser點擊Get請求(我使用VS code插件來啟動Server) 我自己用’Live Server’
   4. 可看到get成功! 也可以看到因為是GET請求所以他把參數帶到url的’?’後了!
   5. 
   6. 也可以加上:
5. <div><span>{{msg}}</span></div> <!-- 將下方請求的log印出來! -->
   * 1. 這樣在UI上就可以看到res.data被印出來顯示在上方了!
     2. 

如果試著把$this.http.get改為不存在的位址: 像是改為

1. this.$http.get("package2.json", { // 可以試著改為一個不存在的，例如package1.json，就會報錯!
   * 1. 就會噴錯404啦!!!
     2. 

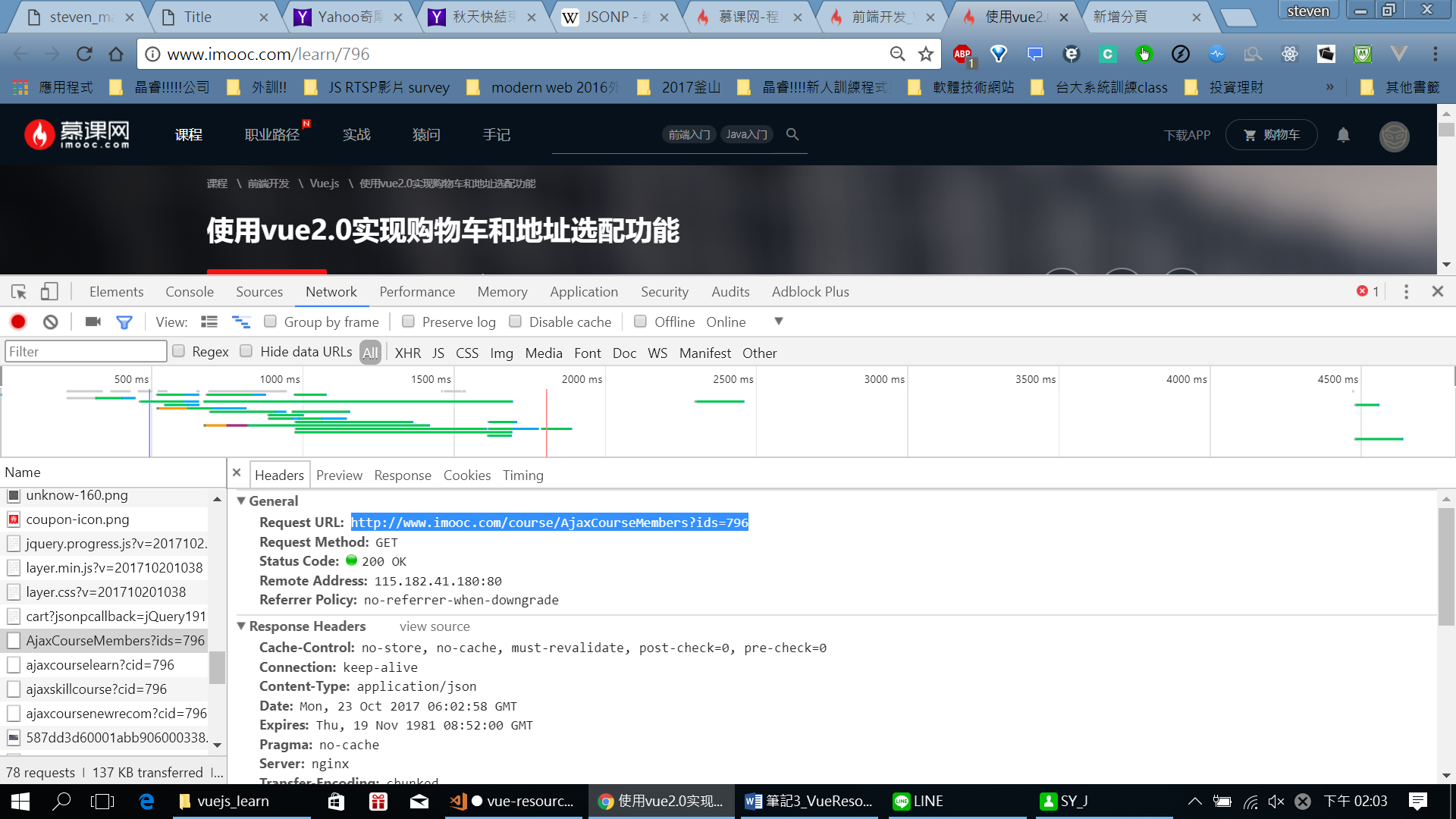
<POST>: 

但Live Server(VS code套件)好像不給發POST.. 待確認

<JSONP 請求> 跨站點請求:

<JSONP模擬我們的script標籤來發送GET請求>

首先假設用此url



將此url放到剛剛寫得的GET測試

會發現報錯了!



>> 不允許跨域訪問!!! (跨域訪問是不允許的!)

JSONP[[編輯](https://zh.wikipedia.org/w/index.php?title=JSONP&action=edit&section=0&summary=/*%20%E9%A6%96%E6%AE%B5%20*/%20)]

維基百科，自由的百科全書

**JSONP**（**JSON with Padding**）是資料格式[JSON](https://zh.wikipedia.org/wiki/JSON" \o "JSON)的一種「使用模式」，可以讓網頁從別的網域要資料。另一個解決這個問題的新方法是[跨來源資源共享](https://zh.wikipedia.org/wiki/%E8%B7%A8%E4%BE%86%E6%BA%90%E8%B3%87%E6%BA%90%E5%85%B1%E4%BA%AB" \o "跨來源資源共享)。

由於[同源策略](https://zh.wikipedia.org/w/index.php?title=%E5%90%8C%E6%BA%90%E7%AD%96%E7%95%A5&action=edit&redlink=1)，一般來說位於server1.example.com的網頁無法與不是 server1.example.com的伺服器溝通，而[HTML](https://zh.wikipedia.org/wiki/HTML" \o "HTML)的 [<script>](https://zh.wikipedia.org/wiki/HTML%E5%85%83%E7%B4%A0#script_tag)元素是一個例外。利用 [<script>](https://zh.wikipedia.org/wiki/HTML%E5%85%83%E7%B4%A0#script_tag)元素的這個開放策略，網頁可以得到從其他來源動態產生的JSON資料，而這種使用模式就是所謂的 JSONP。用JSONP抓到的資料並不是JSON，而是任意的JavaScript，用 JavaScript直譯器執行而不是用JSON解析器解析。

我們來寫jsonp請求吧!>>

新增method吧:

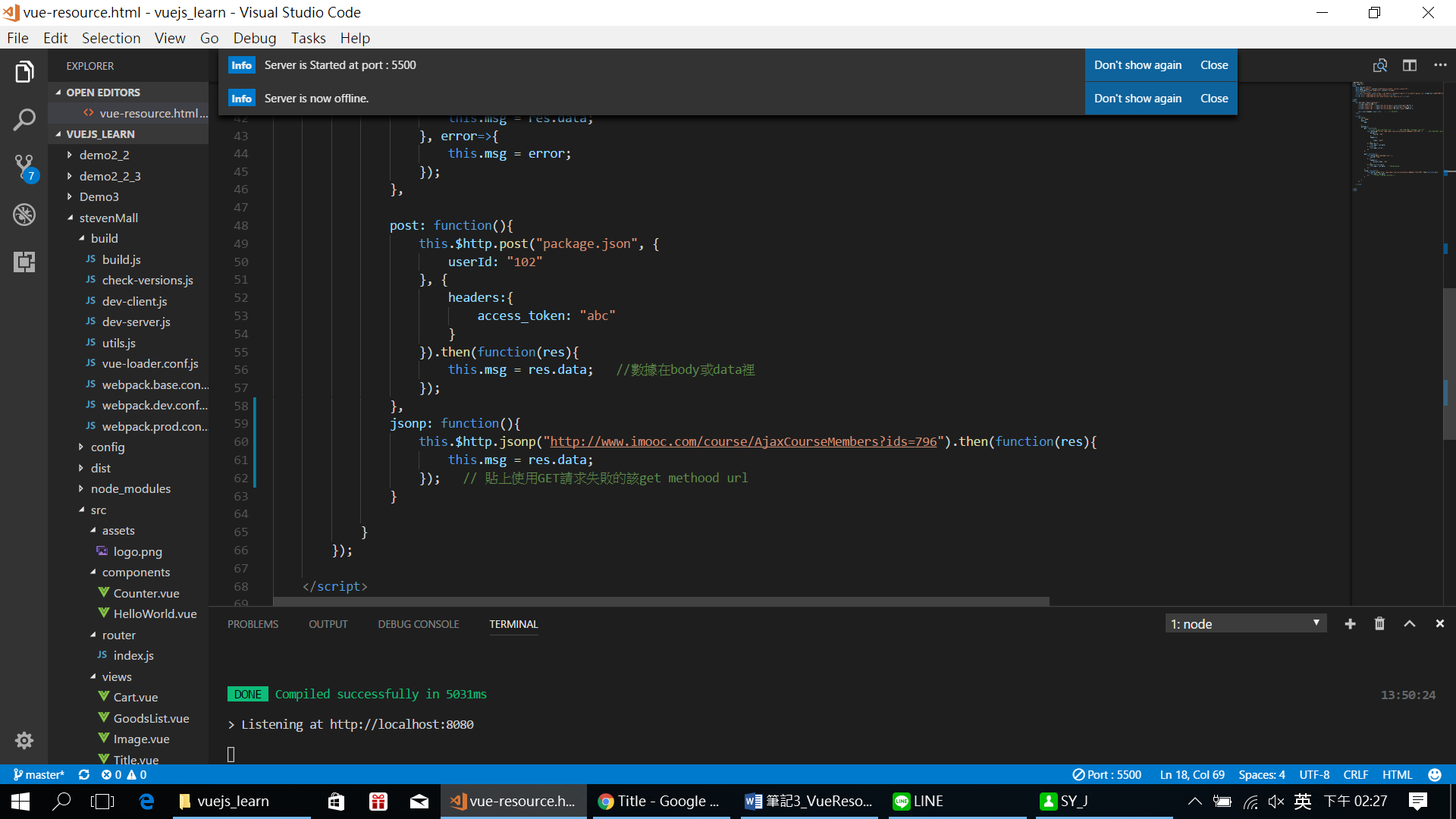
jsonp: function(){

this.$http.jsonp("http://www.imooc.com/course/AjaxCourseMembers?ids=796").then(function(res){

this.msg = res.data;

}); // 貼上使用GET請求失敗的該get methood url

}



可以看到成功囉>>



7. 接下來是說明全局攔截器(interceptor)是如何使用的

// 了解全局攔截器的使用:

mounted: function(){

Vue.http.interceptors.push(function(request, next){

console.log("request init."); // 即發送請求前會優先進到這裡優先執行!

next(function(response){ // 下一步代表已經響應了

console.log("response init.")

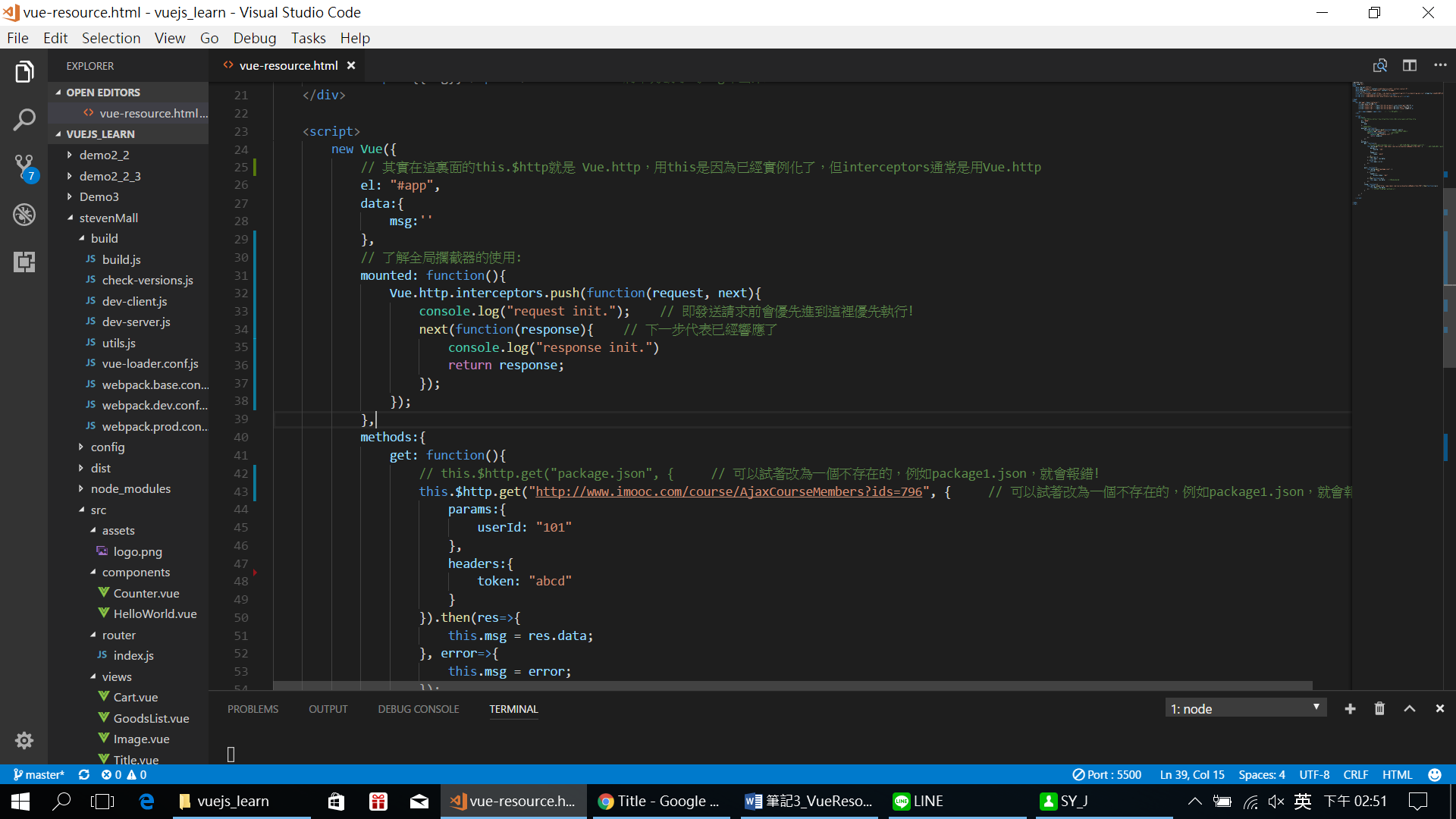
return response;

});

});

},

新增mounted:



可以看到: 如果用GET發送剛不能直接get的url:



1. Http插件(可以寫上root路徑，如此get post等只要寫上檔案名字!):

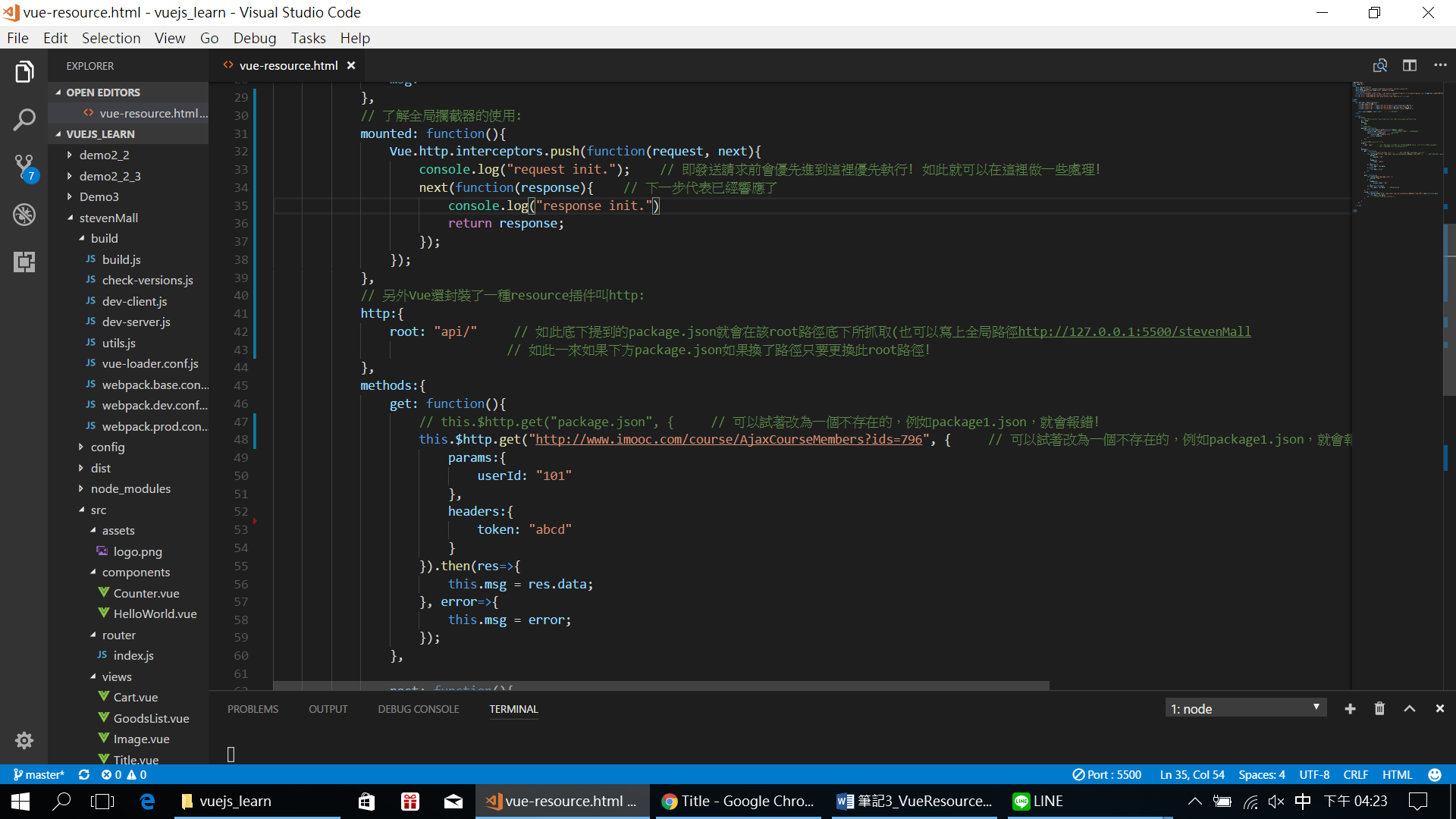
// 另外Vue還封裝了一種resource插件叫http:

http:{

root: "api/" // 如此底下提到的package.json就會在該root路徑底下所抓取(也可以寫上全局路徑http://127.0.0.1:5500/stevenMall

// 如此一來如果下方package.json如果換了路徑只要更換此root路徑!，如此get post等只要寫上檔案名字

},



另外還可以用這種方式:

http: function(){ // 也可以用http 指定傳參數、指定方法的方式

this.$http({

url: "package.json",

methods: 'GET',

params: {

userId: "103"

},

headers:{

token: "123"

},

timeout: 50,

before: function(){

console.log("Before init")

}

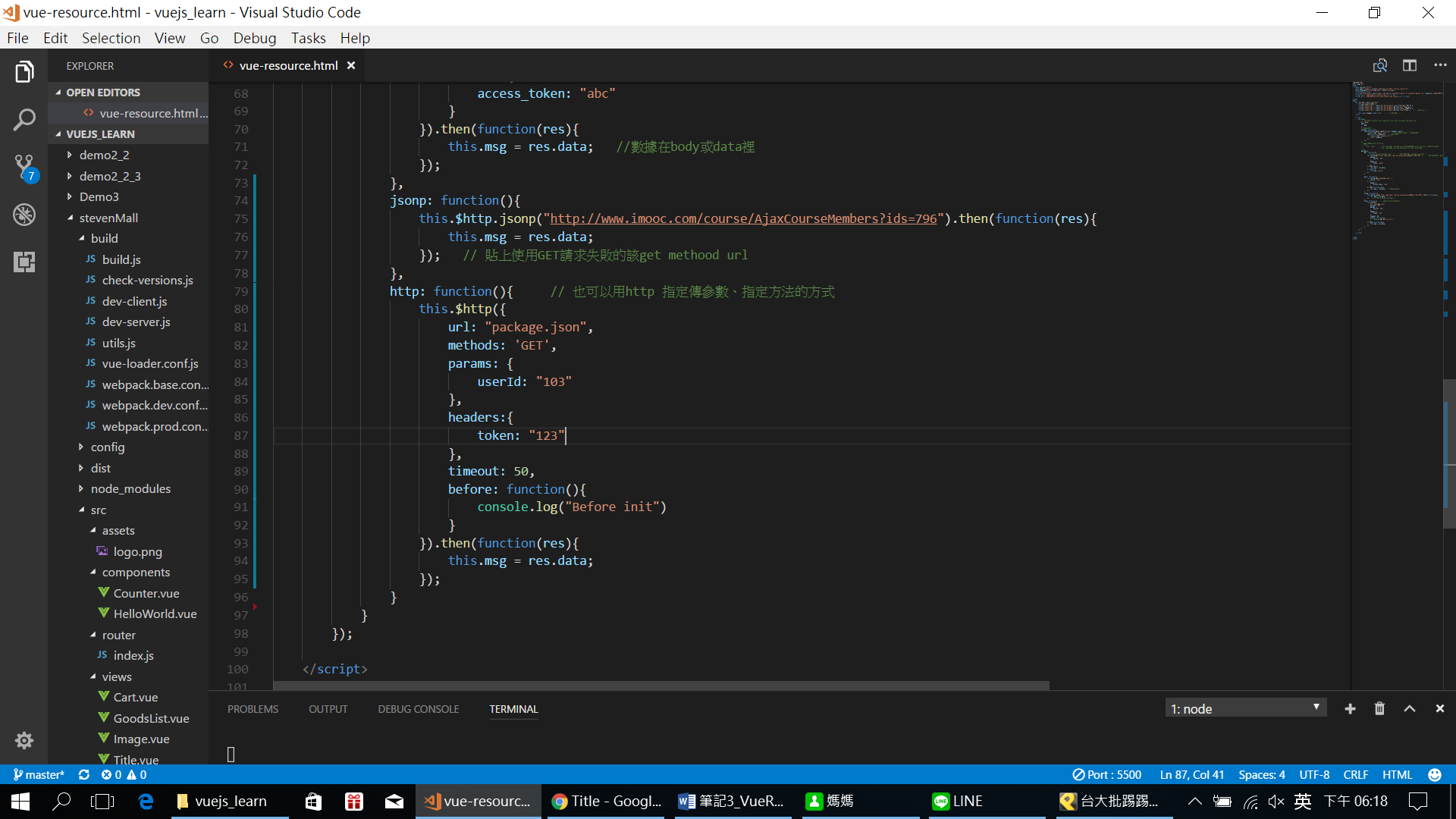
}).then(function(res){

this.msg = res.data;

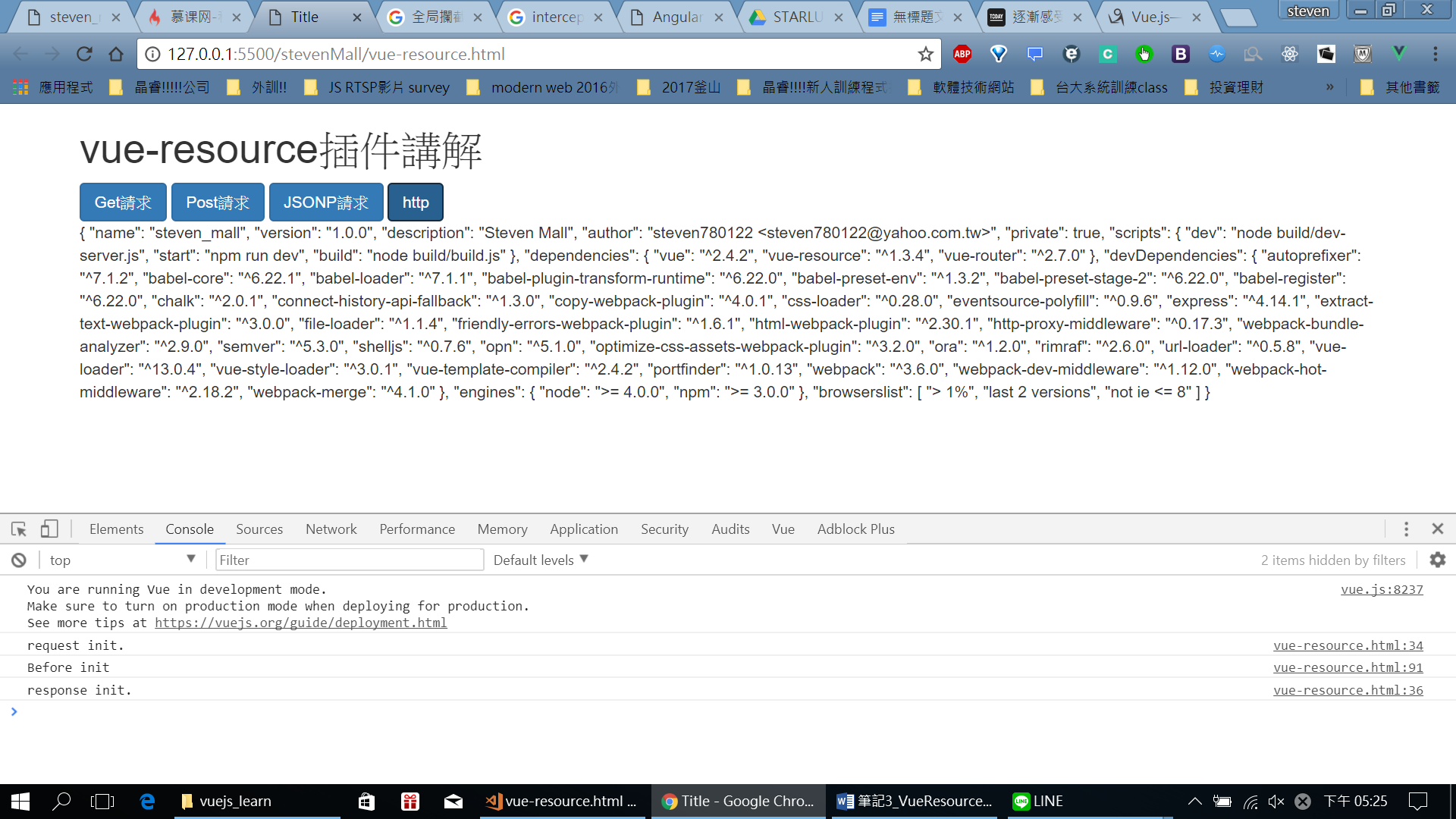
});

}

來指定method



結果:



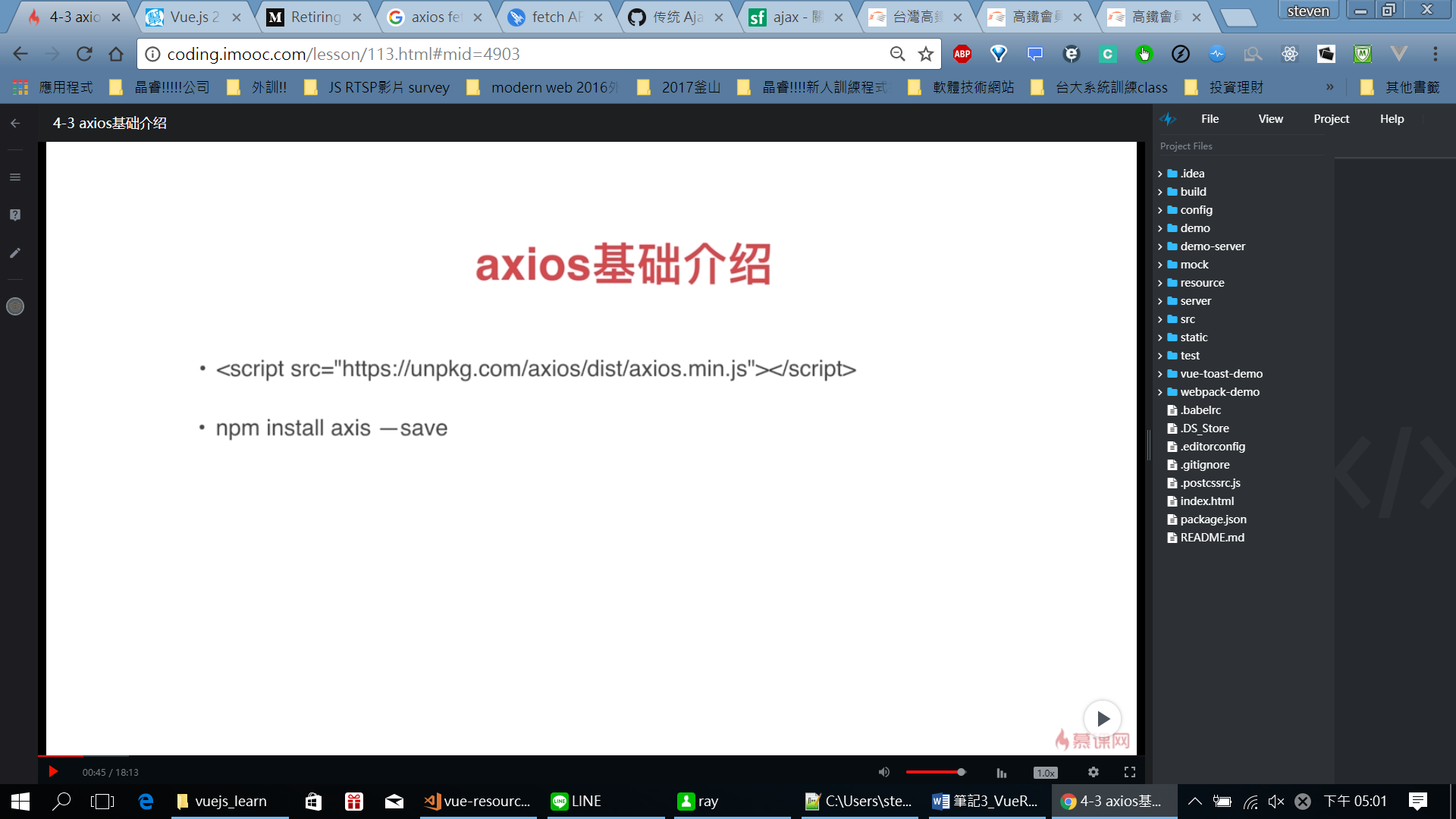
<>

這都類似於Jquery 的AJAX插件

<Axios> 2016年底作者Vue-Resource不再是官方推薦，推薦使用Axios!!

(Hubert: 提到現在應該還是fetch API(比較底層)一統)

安裝:





會發現沒jsonP !! 多了一個options (類似前面寫方法AJAX直接在大括號進行配置)

多了.all 同時把getUserAccount和getUserPermissions同時傳到array(數組)內，再透過spread函數進行展開，可以同時獲得這2個街口!!

(傳統的方式就是先獲取第一個接口，再用他的回呼函式去調用第2個接口)

演示: