

Vue實務案例：標籤選單，整合多個資料串接

1. 實務開發目標：在網頁上方做一個「標籤選單」，這個標籤選單是可以切換成 產品資料 及 評論訊息，一開始 使用者是可以看到 「產品資料」，這個產品資料是串接後端提供的網址，使用者可以點選評論訊息，那這個標籤選單 就可以告訴使用者 現在正在看評論訊息，「評論訊息」標籤的樣式會有變化，變成粗體字，產品資料就變成一般體，同時底下的內容就要串接另一個後端網址，得到評論訊息畫在網頁上：



2. 組件架構及對於響應式狀態的設計，最根部的組件是 App.vue，App.vue 會管理整理最上層的概念，就是標籤選單的切換，切換到產品資料，我們就稱之為產品資料的頁面，切換到評論訊息，就稱之為「評論訊息的頁面」，然後利用響應式狀態（page）記錄今天使用者在看「產品頁面」還是「評論訊息的頁面」，如果使用者在看「產品頁面」ProductPage，我們就把產品頁的組件畫出來，在產品頁面的組件中，維護一個響應式狀態叫 Products，用來記錄我們從後端抓到的產品資料，在產品頁面的底下，另外做一個組件叫 ProductList，用來決定怎麼樣呈現這個響應式狀態的資料，怎麼樣把它畫到畫面上，如果使用者切換到評論訊息的頁面，我們就會改變這個 page，就會載入另一個叫 ReviewPage，這個評論訊息的頁面就會去抓後端提供的評論訊息，管理一個 reviews 響應式狀態，整個結構跟產品結構是非常類似的。底下一樣做一個 ReviewList 用來呈現這些評論資料。

YouTube 首頁

實務開發目標

組件架構和狀態設計

```
graph TD; App[App<br/>page] --> ProductPage[ProductPage<br/>products]; App --> ReviewPage[ReviewPage<br/>reviews]; ProductPage --> ProductList[ProductList]; ReviewPage --> ReviewList[ReviewList];
```

3. 開發準備事項：

後端提供的資料網址:

產品資料

https://lilylovesjeans.github.io/Learning_Vue/Day13-Vue實務案例_標籤選單_整合多個資料串接/products.json

評論訊息

https://lilylovesjeans.github.io/Learning_Vue/Day13-Vue實務案例_標籤選單_整合多個資料串接/reviews.json

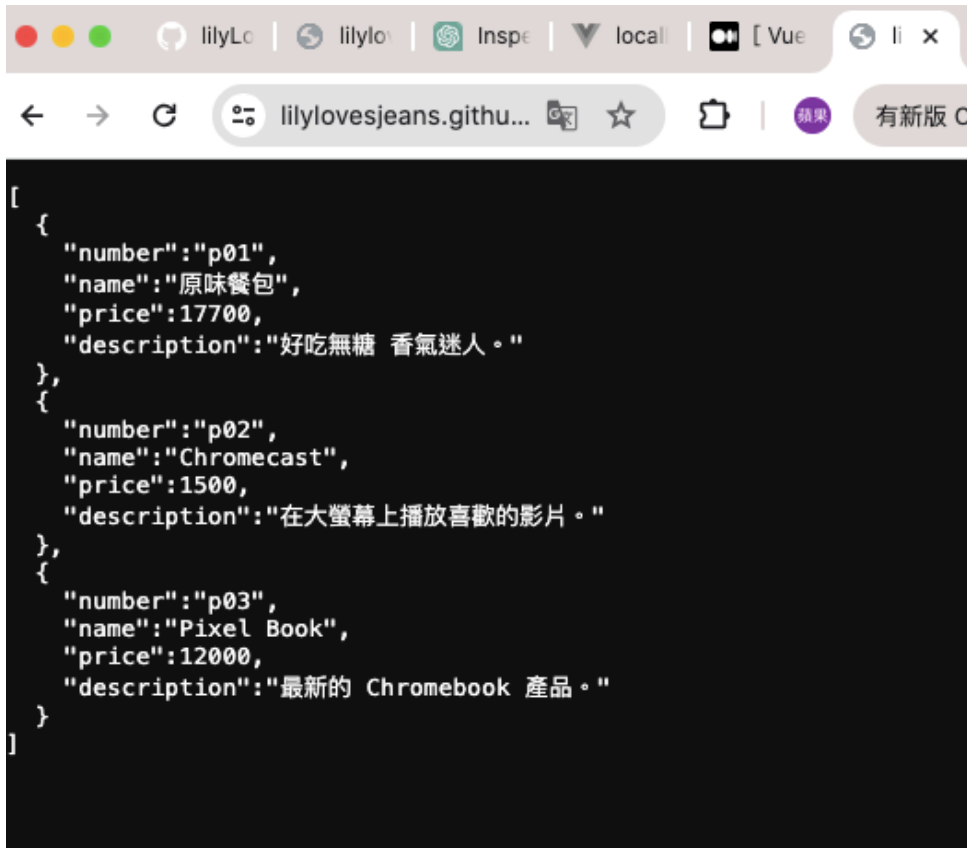
實務開發目標

開發準備事項

- 後端提供的資料網址
 - <https://cwpeng.github.io/live-rsamples/data/products.json>
 - <https://cwpeng.github.io/live-rsamples/data/reviews.json>
- JavaScript fetch 基本用法
- Vue 樣板語法的流程控制
- Vue 響應式狀態、生命週期
- Vue 組件切割操作

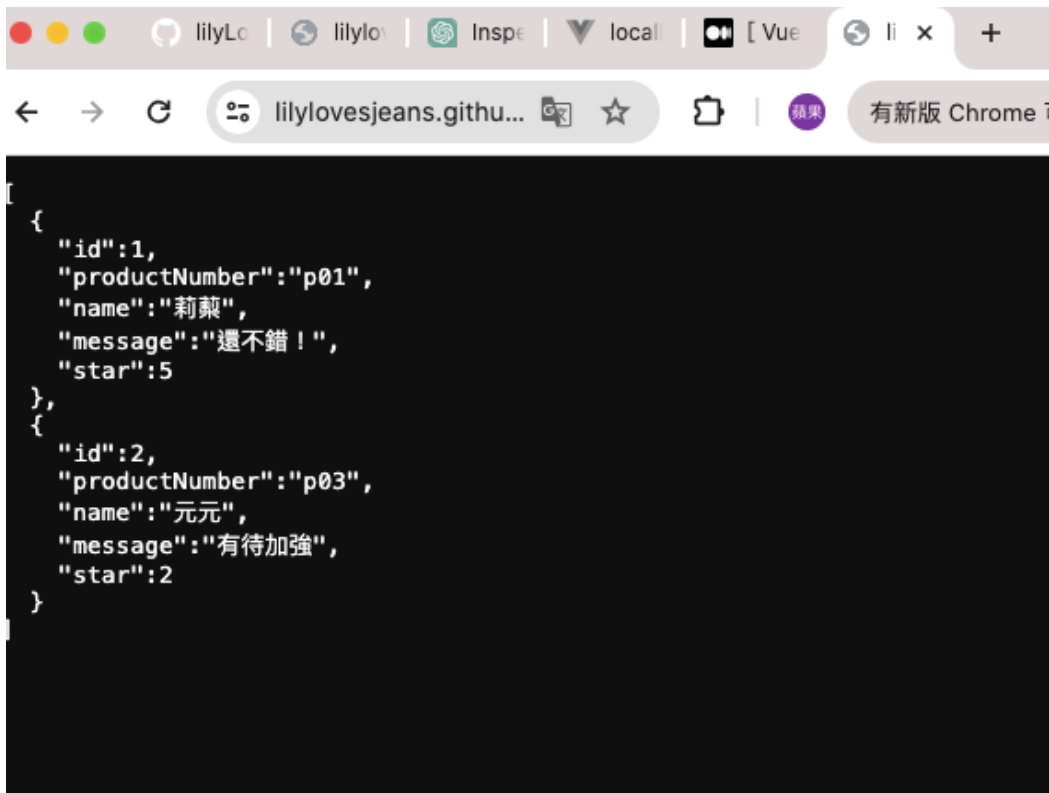
4. 開發開始：

先打開網址，看是否連的上：



A screenshot of a web browser window. The address bar shows the URL "lilylovesjeans.github...". The page content is a JSON array of three product objects displayed in a dark-themed code editor. The first object has "number": "p01", "name": "原味餐包", "price": 17700, and "description": "好吃無糖 香氣迷人。". The second object has "number": "p02", "name": "Chromecast", "price": 1500, and "description": "在大螢幕上播放喜歡的影片。". The third object has "number": "p03", "name": "Pixel Book", "price": 12000, and "description": "最新的 Chromebook 產品。".

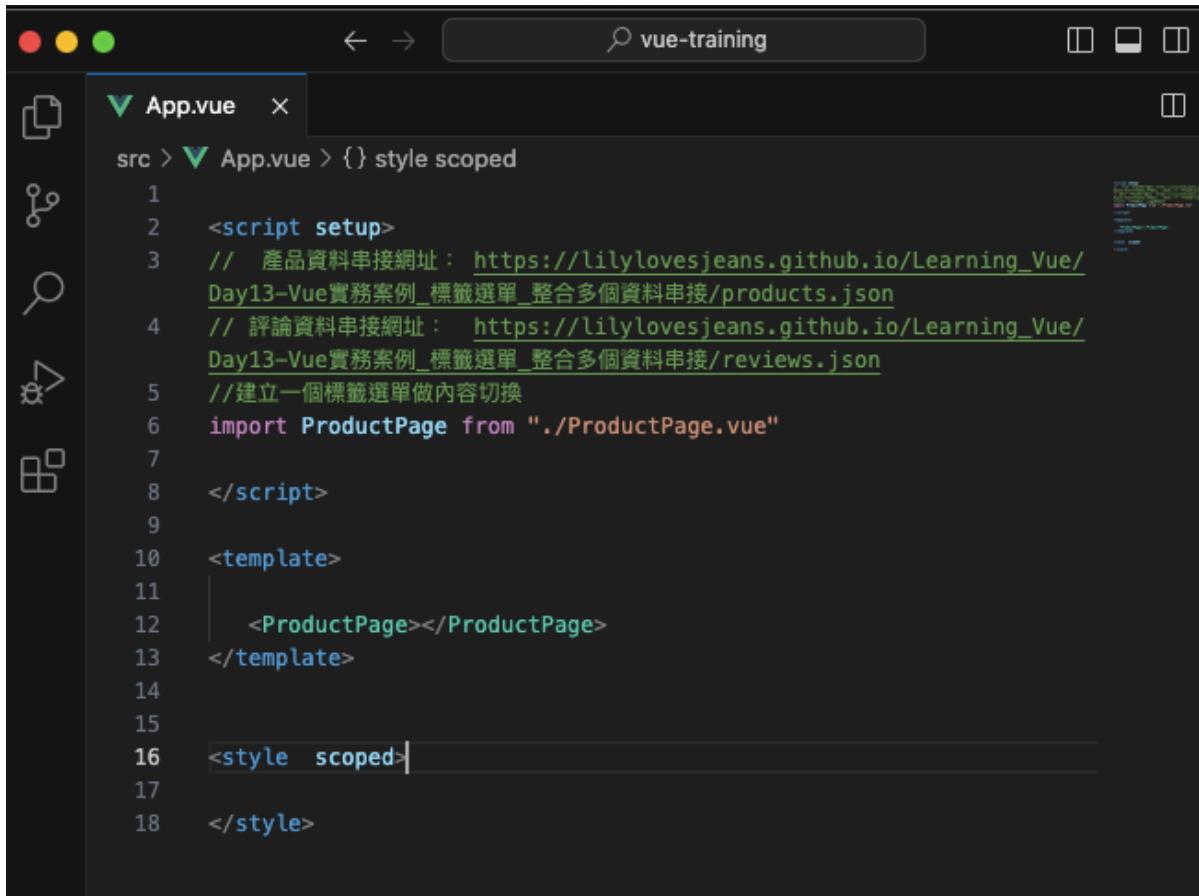
```
[
  {
    "number": "p01",
    "name": "原味餐包",
    "price": 17700,
    "description": "好吃無糖 香氣迷人。"
  },
  {
    "number": "p02",
    "name": "Chromecast",
    "price": 1500,
    "description": "在大螢幕上播放喜歡的影片。"
  },
  {
    "number": "p03",
    "name": "Pixel Book",
    "price": 12000,
    "description": "最新的 Chromebook 產品。"
  }
]
```



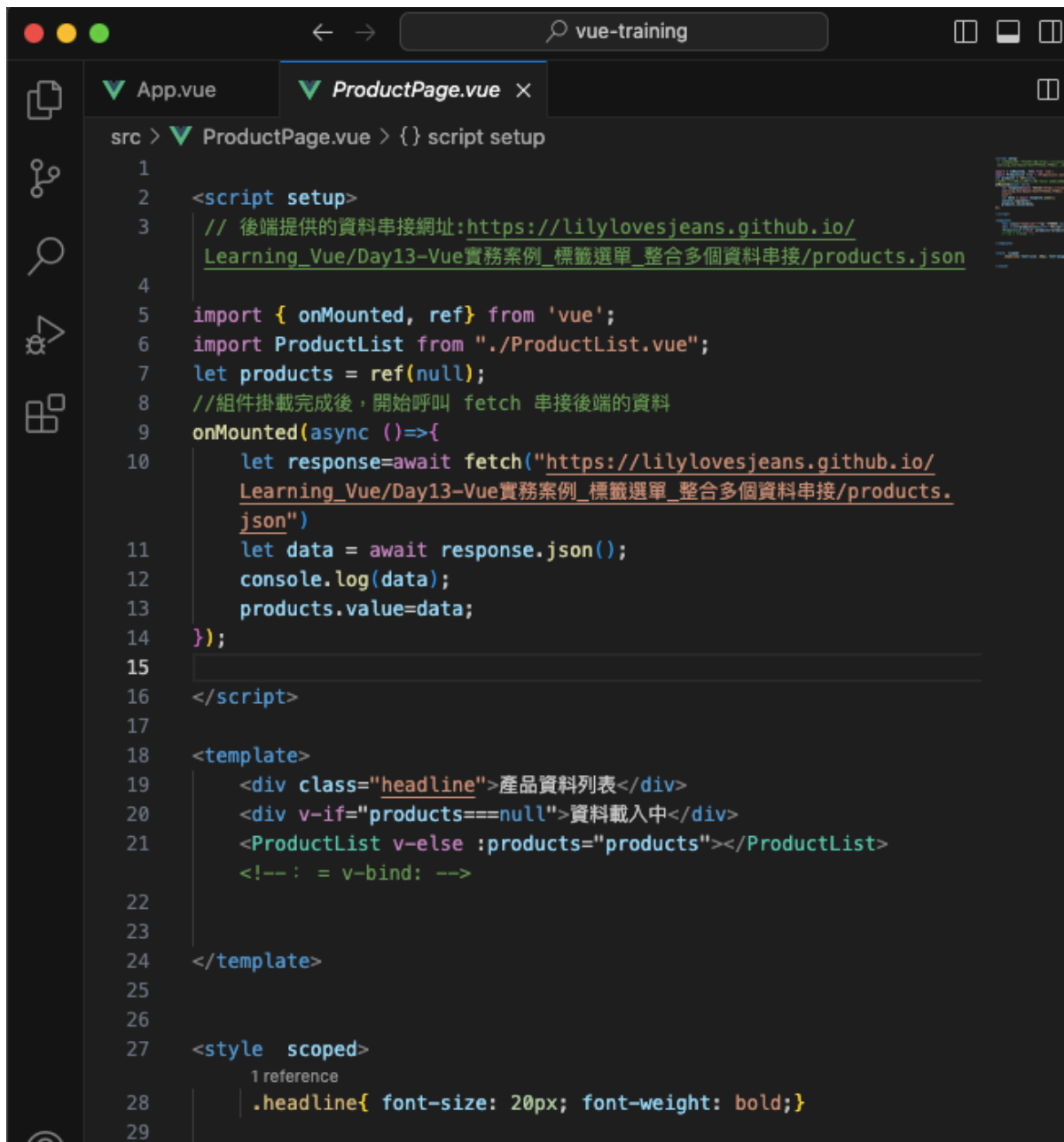
A screenshot of a web browser window. The address bar shows the URL "lilylovesjeans.github...". The page content is a JSON array of two user review objects displayed in a dark-themed code editor. The first object has "id": 1, "productNumber": "p01", "name": "莉蕻", "message": "還不錯！", and "star": 5. The second object has "id": 2, "productNumber": "p03", "name": "元元", "message": "有待加強", and "star": 2.

```
[
  {
    "id": 1,
    "productNumber": "p01",
    "name": "莉蕻",
    "message": "還不錯！",
    "star": 5
  },
  {
    "id": 2,
    "productNumber": "p03",
    "name": "元元",
    "message": "有待加強",
    "star": 2
  }
]
```

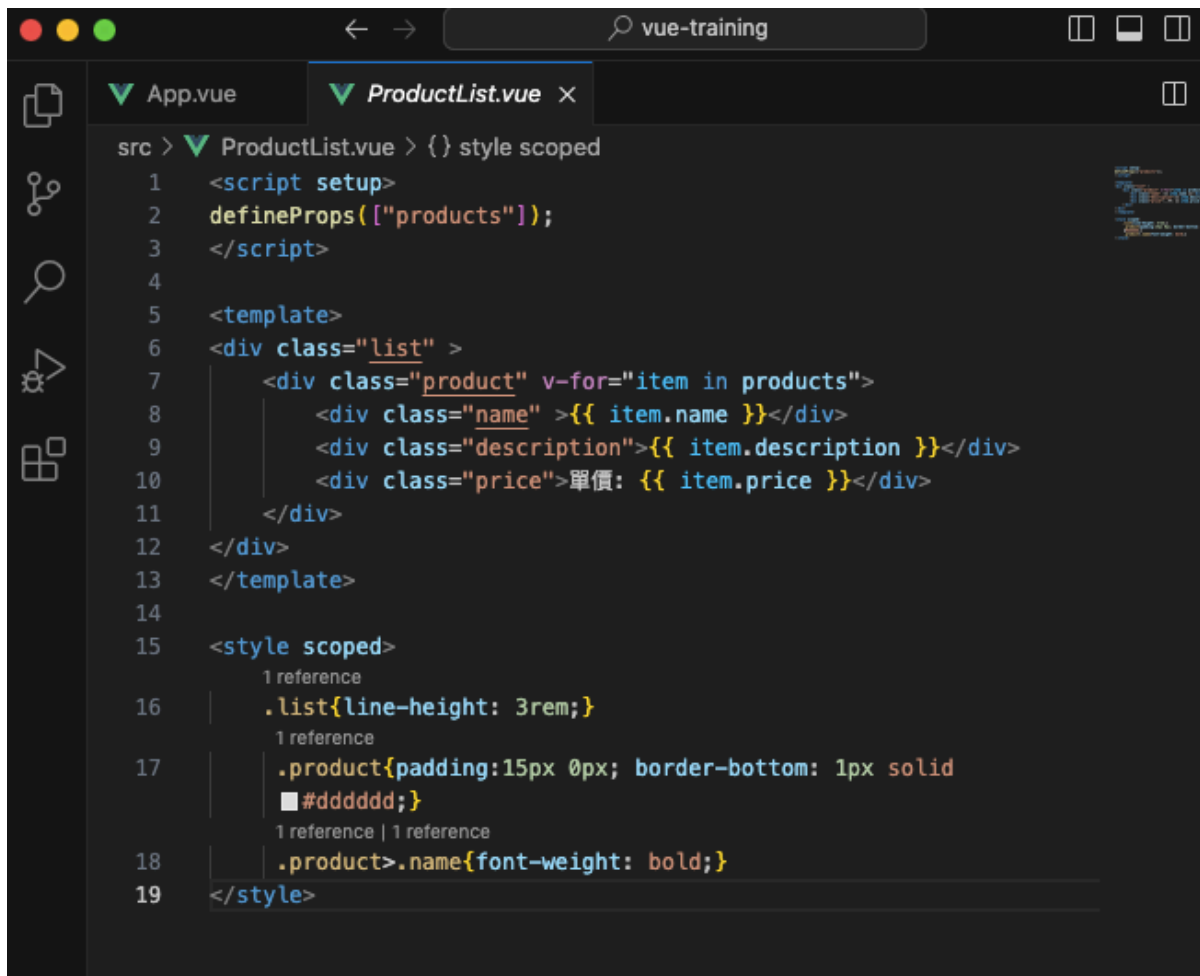
5. 先在 App.vue 載入 ProductPage , 看看是否有抓到資料:



```
src > App.vue > {} style scoped
1
2 <script setup>
3 // 產品資料串接網址: https://lilylovesjeans.github.io/Learning\_Vue/
4 // 評論資料串接網址: https://lilylovesjeans.github.io/Learning\_Vue/
5 // 建立一個標籤選單做內容切換
6 import ProductPage from "./ProductPage.vue"
7
8 </script>
9
10 <template>
11   <ProductPage></ProductPage>
12 </template>
13
14
15
16 <style scoped>
17
18 </style>
```



```
src > ProductPage.vue > {} script setup
1
2 <script setup>
3   // 後端提供的資料串接網址:https://lilylovesjeans.github.io/
   Learning_Vue/Day13-Vue實務案例_標籤選單_整合多個資料串接/products.json
4
5   import { onMounted, ref } from 'vue';
6   import ProductList from "../ProductList.vue";
7   let products = ref(null);
8   //組件掛載完成後，開始呼叫 fetch 串接後端的資料
9   onMounted(async ()=>{
10     let response=await fetch("https://lilylovesjeans.github.io/
       Learning_Vue/Day13-Vue實務案例_標籤選單_整合多個資料串接/products.
       json")
11     let data = await response.json();
12     console.log(data);
13     products.value=data;
14   });
15
16 </script>
17
18 <template>
19   <div class="headline">產品資料列表</div>
20   <div v-if="products===null">資料載入中</div>
21   <ProductList v-else :products="products"></ProductList>
22   <!-- : = v-bind: -->
23
24 </template>
25
26
27 <style scoped>
28   1 reference
29   .headline{ font-size: 20px; font-weight: bold;}
```



```
src > ProductList.vue > {} style scoped
1  <script setup>
2  defineProps(["products"]);
3  </script>
4
5  <template>
6  <div class="list" >
7    <div class="product" v-for="item in products">
8      <div class="name" >{{ item.name }}</div>
9      <div class="description">{{ item.description }}</div>
10     <div class="price">單價: {{ item.price }}</div>
11   </div>
12 </div>
13 </template>
14
15 <style scoped>
16   .list{line-height: 3rem;}
17   .product{padding:15px 0px; border-bottom: 1px solid #dddddd;}
18   .product>.name{font-weight: bold;}
19 </style>
```

6. 開始寫：
在 App.vue

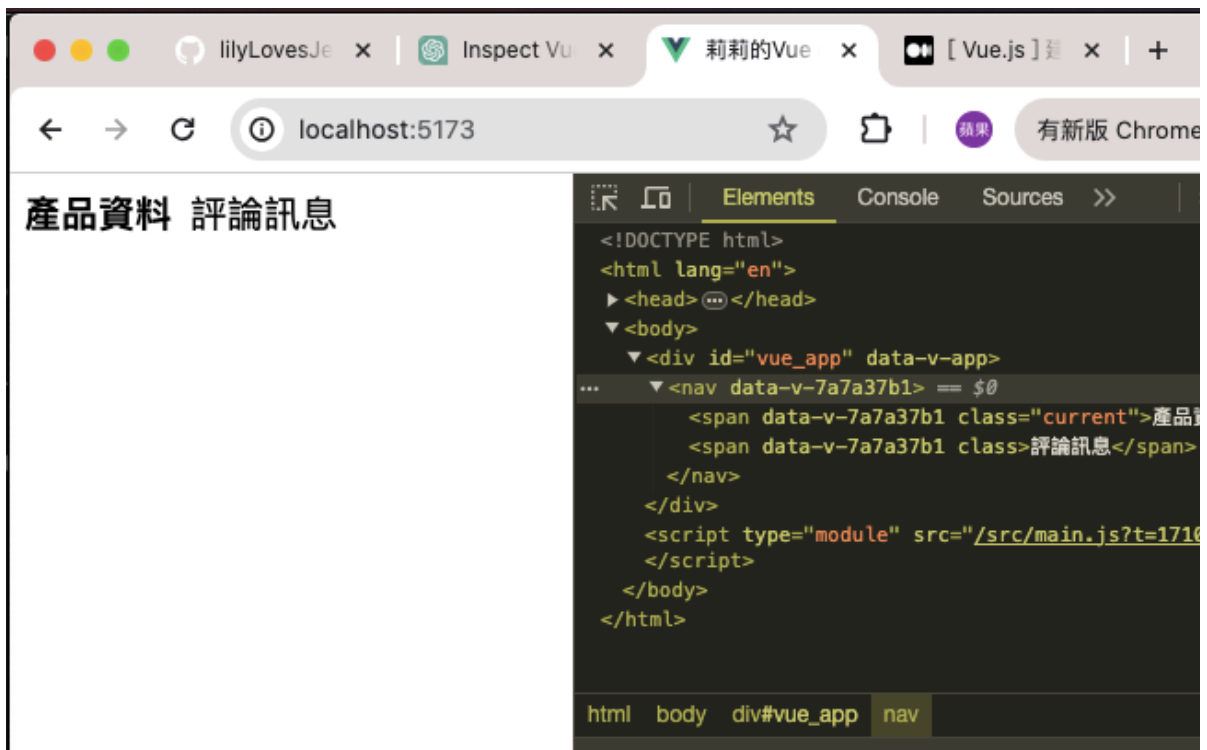
```
//記錄一個 page的響應式狀態，用來決定現在要看的是哪一個內容，product 或者是 review，預設定 product
```

1. 在導覽列做二個 span 產品資料 及 評論訊息，
先給它預設簡單樣式，預設文字大小 20px，span 接一些距離，每個標籤靠左，向右拉10個像素。
這二個span 要怎麼做切換的概念？先做一個樣式的處理，在 style: nav>span 是被選到的page，就給予粗體字：
這樣表達: nav>span的類別 如果是 current，就給予粗體字: nav>span.current{font-weight: bold}，

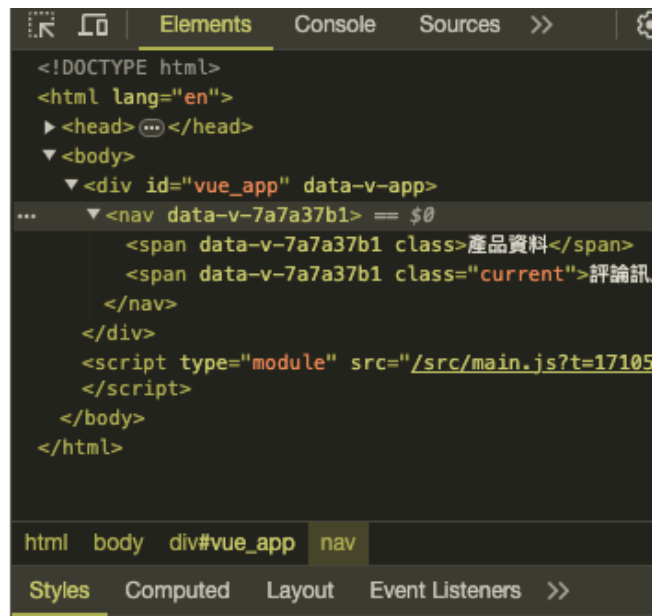
```
vue-training

App.vue x ProductList.vue

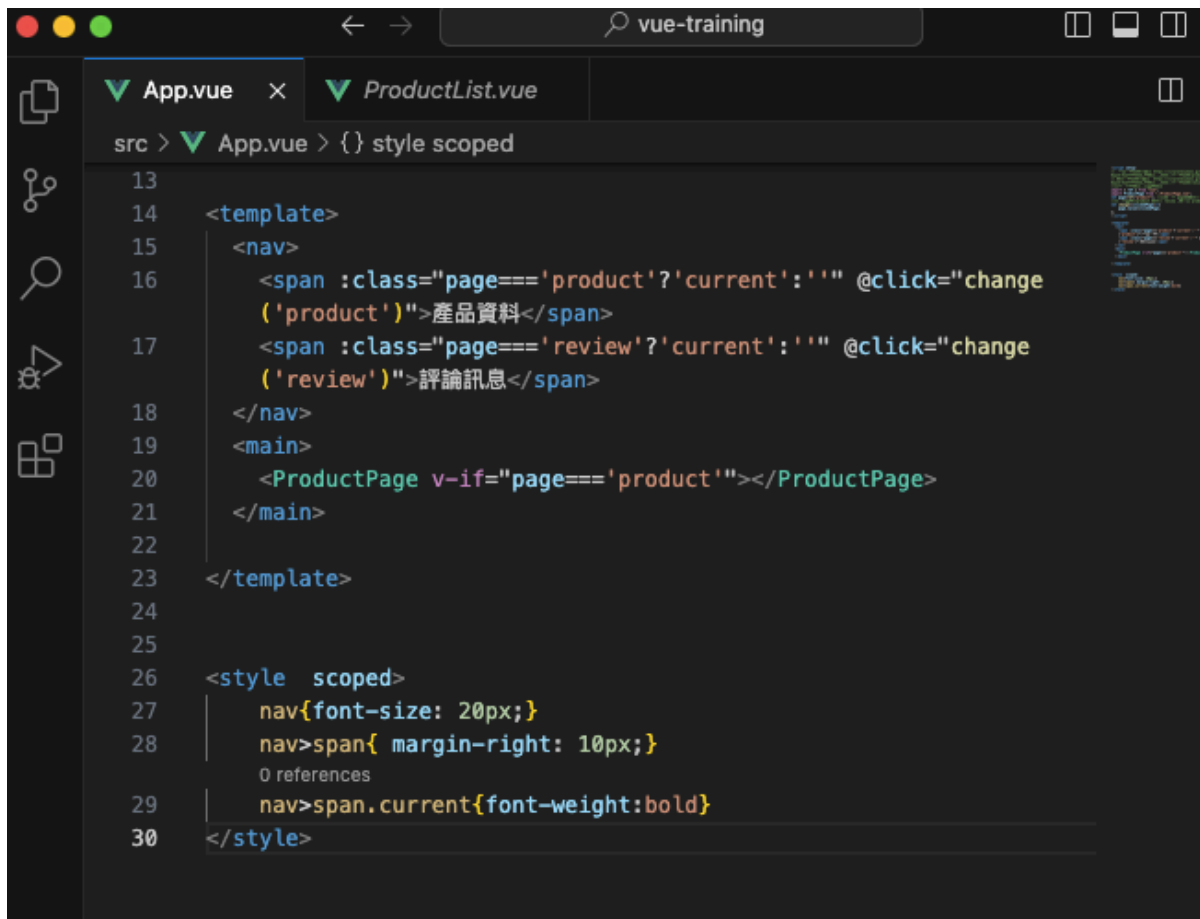
src > App.vue > {} template
1
2 <script setup>
3 // 產品資料串接網址: https://lilylovesjeans.github.io/Learning_Vue/
  Day13-Vue實務案例_標籤選單_整合多個資料串接/products.json
4 // 評論資料串接網址: https://lilylovesjeans.github.io/Learning_Vue/
  Day13-Vue實務案例_標籤選單_整合多個資料串接/reviews.json
5 //建立一個標籤選單做內容切換
6 import { ref } from "vue";
7 let page=ref("product"); //記錄一個 page的響應式狀態，用來決定現在要看的
  是哪一個內容，product 或者是 review ,預設定 product
8 let change=(clickedPage)=>{
9   page.value=clickedPage;
10 };
11 </script>
12
13 <template>
14   <nav>
15     <span :class="page==='product'? 'current': ''" @click="change
      ('product')">產品資料</span>
16     <span :class="page==='review'? 'current': ''" @click="change
      ('review')">評論訊息</span>
17   </nav>
18
19 </template>
20
21
22 <style scoped>
23   nav{font-size: 20px;}
24   nav>span{ margin-right: 10px;}
25   nav>span.current{font-weight:bold}
26 </style>
```



產品資料 評論訊息



7. 如果使用者點擊產品資料，就要在下面顯元出產品資料的內容，在<nav>下列新增一個 <main> 載入 ProductPage 組件，然後畫出來，並判斷現在點的是 product 才能 show出來：



```
13
14 <template>
15   <nav>
16     <span :class="page==='product'? 'current': ''" @click="change
17       ('product')">產品資料</span>
18     <span :class="page==='review'? 'current': ''" @click="change
19       ('review')">評論訊息</span>
20   </nav>
21   <main>
22     <ProductPage v-if="page==='product'"></ProductPage>
23   </main>
24 </template>
25
26 <style scoped>
27   nav{font-size: 20px;}
28   nav>span{ margin-right: 10px;}
29   0 references
30   nav>span.current{font-weight:bold}
31 </style>
```

產品資料 評論訊息

產品資料列表

原味餐包

好吃無糖 香氣迷人。

單價: 17700

Chromecast

在大螢幕上播放喜歡的影片。

單價: 1500

Pixel Book

最新的 Chromebook 產品。

單價: 12000

Elements

<!DOCTYPE html>

<html lang="en">

<head>...</head>

<body>

<div id="vue_app" data-v-7a7a37b1>

<nav data-v-7a7a37b1>

<main data-v-7a7a37b1>

</div>

<script type="module" src="595869942"></script>

</body>

</html>

html body div#vue_app

Styles

Computed

Layout

Filter

:hov .cls

+

⌵

element.style {

}

user agent stylesheet

div {

display: block;

}

Console

Search

W

⌵

⌵

top

⌵

⌵

Filter

All levels

No Issues

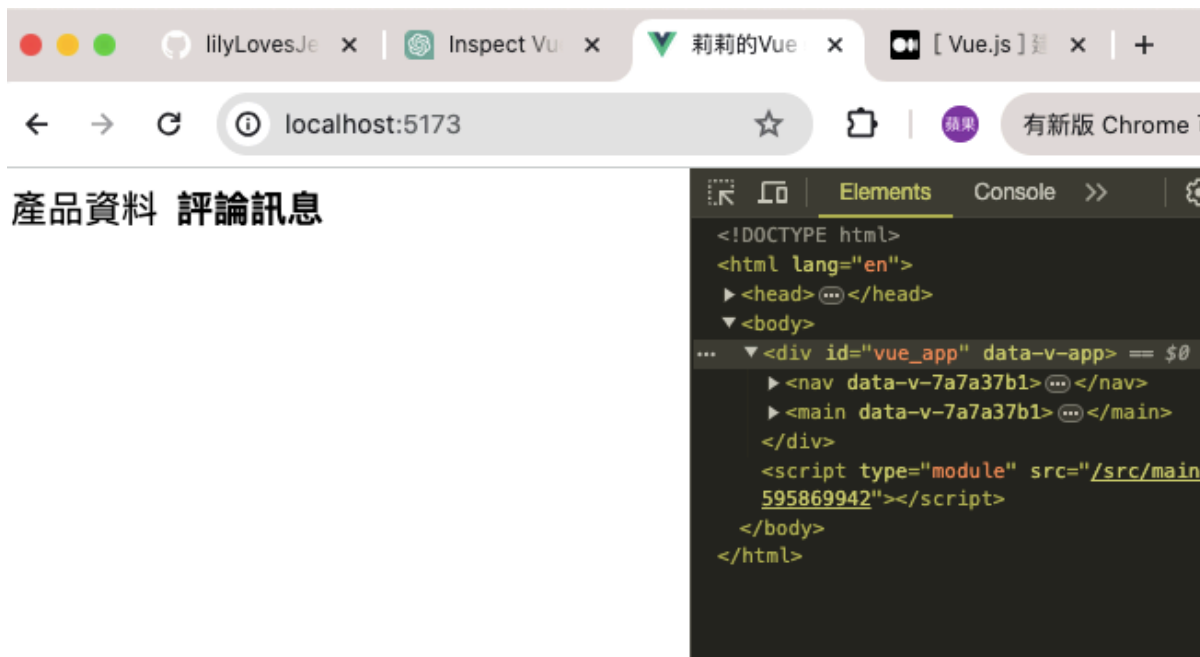
6 h

▶

⌵

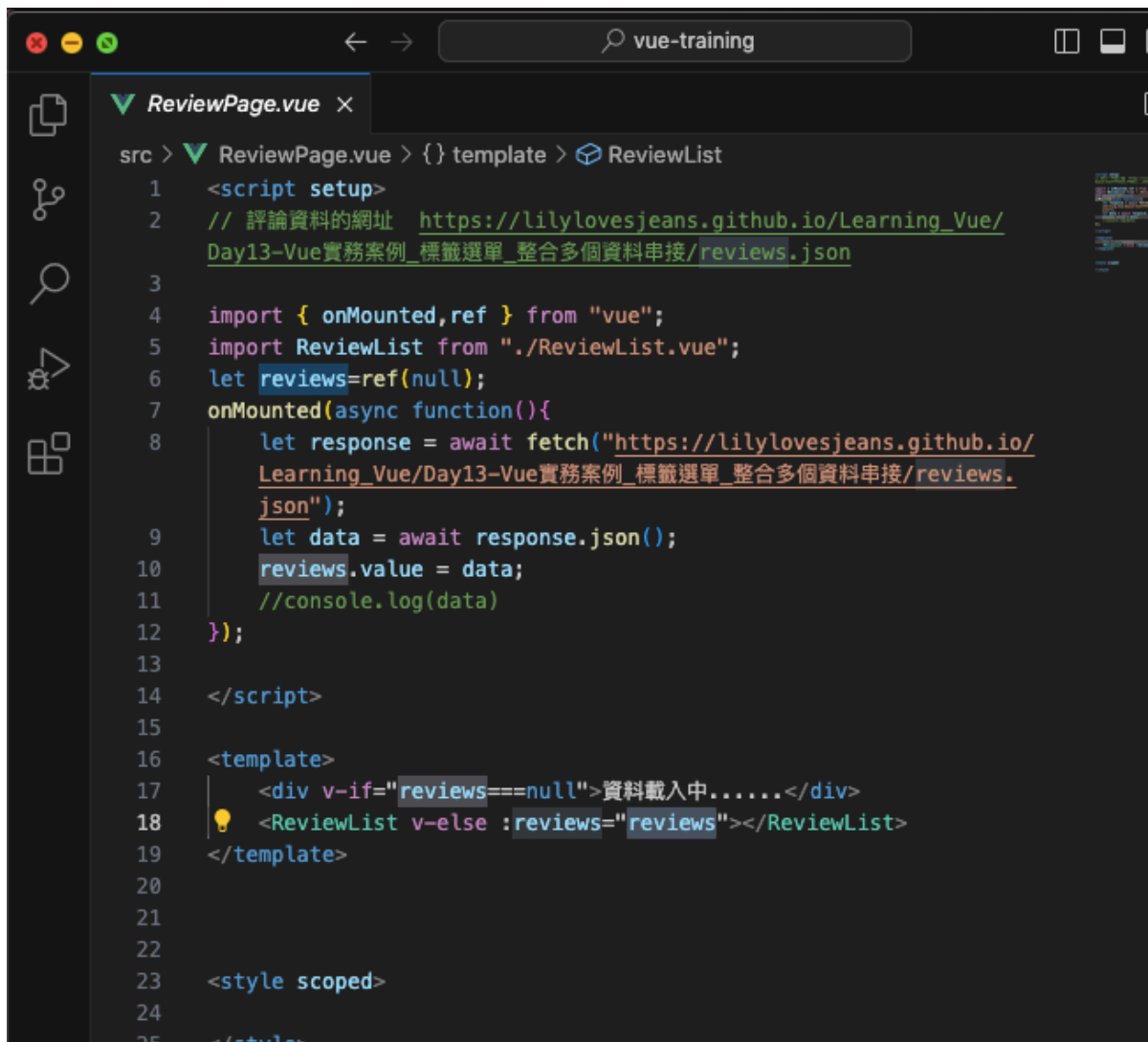
6 messages

>



8. 把評論資料也show出來，新建一個 ReviewPage.vue：

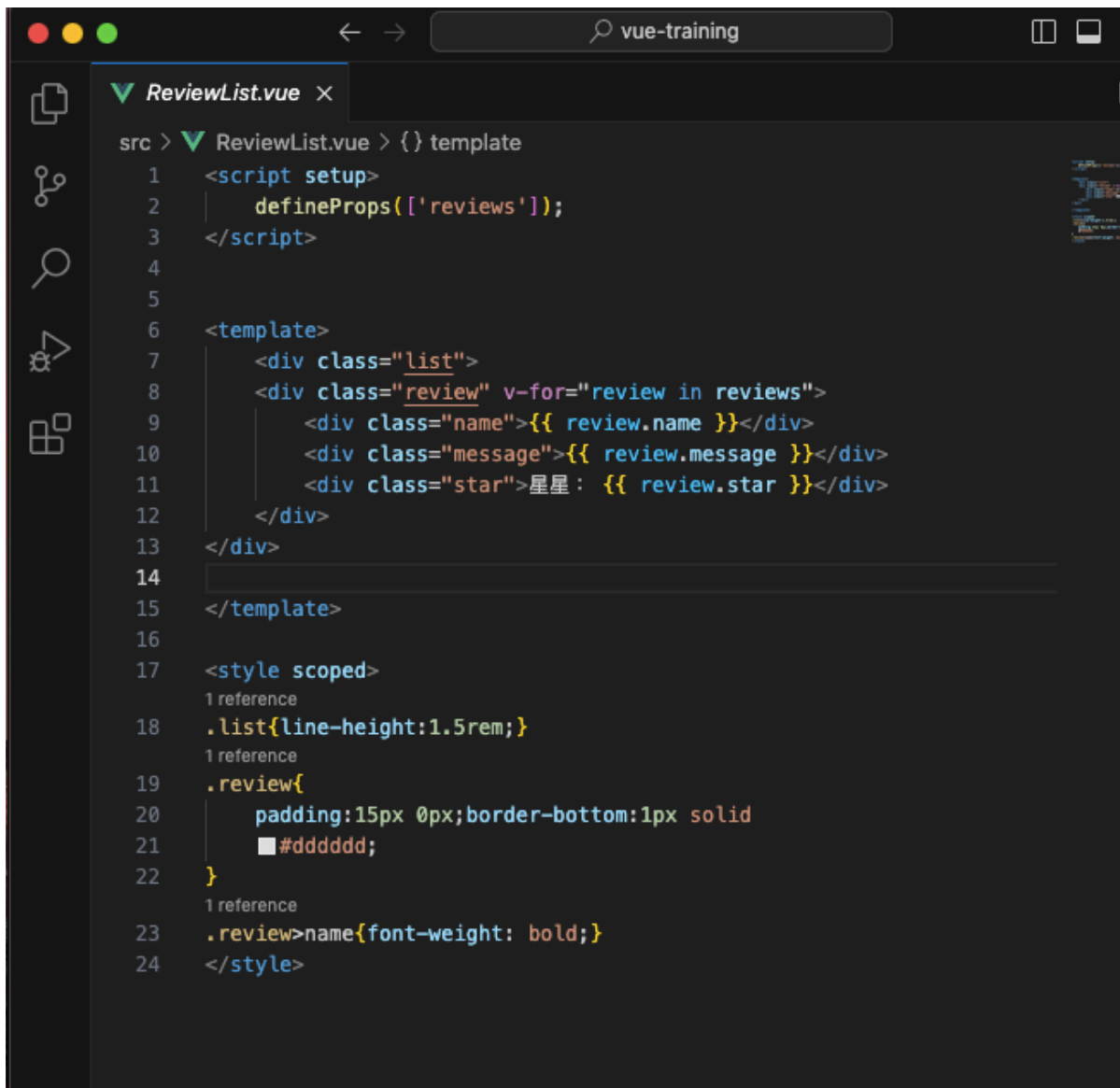
因為要跟後端串接網址，需要一個響應式狀態及掛載的時機點，所以需要寫vue 載入 ref 及 onMounted，建立一個響應式狀態 reviews，記錄後端評論資料，default 為 null，然後當掛載之後，要去抓取後端評論資料，使用 fetch 抓取資料，抓取資料後用 json 解讀，



```
src > ReviewPage.vue > {} template > ReviewList
1  <script setup>
2  // 評論資料的網址 https://lilylovesjeans.github.io/Learning_Vue/
   Day13-Vue實務案例_標籤選單_整合多個資料串接/reviews.json
3
4  import { onMounted, ref } from "vue";
5  import ReviewList from "../ReviewList.vue";
6  let reviews = ref(null);
7  onMounted(async function() {
8    let response = await fetch("https://lilylovesjeans.github.io/
   Learning_Vue/Day13-Vue實務案例_標籤選單_整合多個資料串接/reviews.
   json");
9    let data = await response.json();
10   reviews.value = data;
11   //console.log(data)
12 });
13
14 </script>
15
16 <template>
17   <div v-if="reviews===null">資料載入中.....</div>
18   ⚡ <ReviewList v-else :reviews="reviews"></ReviewList>
19 </template>
20
21
22
23 <style scoped>
24
25 </style>
```

然後 新增一個 ReviewList 組件，把抓到的資料畫出來，要把資料從 ReviewPage 傳遞到 ReviewList，則屬性名稱前面要加冒號 像這樣 (`:reviews=reviews`) 動態式響應資料，ReviewList.vue 的 setup 裏要 defineProps，接收上層傳來的資料：`defineProps(["reviews"])`；

要把review 包在list 的標籤裏面，然後簡單用個樣式



```
src > ReviewList.vue > {} template
1  <script setup>
2    |   defineProps(['reviews']);
3  </script>
4
5
6  <template>
7    <div class="list">
8      <div class="review" v-for="review in reviews">
9        <div class="name">{{ review.name }}</div>
10       <div class="message">{{ review.message }}</div>
11       <div class="star">星星: {{ review.star }}</div>
12     </div>
13   </div>
14
15 </template>
16
17 <style scoped>
18   1 reference
19   .list{line-height:1.5rem;}
20   1 reference
21   .review{
22     padding:15px 0px;border-bottom:1px solid
23     #dddddd;
24   }
25   1 reference
26   .review>name{font-weight: bold;}
27 </style>
```

9. 最後 把資料夾做一個處理：把 ProductPage.vue 及 ProductList.vue 放到 src/product 資料夾底下
把 ReviewPage.vue 及 ReviewList.vue 放到 src/review， 做好分門別類：

