

自動換行 ☐

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>useEffect 基礎介紹</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <meta charset="UTF-8">
8   <script src="./javascripts/vendor.js"></script>
9
10  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
11  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
12  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
13
14  <!-- 非同步讀取資料的套件 => axios -->
15  <script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.27.2/axios.min.js"></script>
16  <!-- 元素的高度與寬度參考 -->
17  <!-- https://www.google.com/search?sca_esv=d4df4a45296c77b3&sca_upv=1&rlz=1C1YTUH_zh-TWTW1013TW1013&sxsrf=ADLYWIJZRTfqdYMY7eCE7XZ5C7LQDHUNaA:1
18 </head>
19
20 <body class="bg-lighter">
21   <div class="container main my-5">
22     <div class="row">
23       <main class="col post">
24         <!-- 章節標題 -->
25         <div class="mb-4">
26           <h1 class="mt-0 text-muted">useEffect 基礎介紹</h1>
27         </div>
28
29         <div id="root"></div>
30
31         <script type="text/babel">
32           const { useState, useEffect, useRef } = React;
33
34           const api = 'https://api.unsplash.com/search/photos/';
35           const apiKey = '0p99NnZvv03vGtqrypmaM-LvsICLJS1hFN11Wl20y2M';
36           // 驗證是否能成功連上api
37           // https://api.unsplash.com/photos/?client_id=YOUR_ACCESS_KEY
38
39
40           // 搜尋列的元件
41           const SearchBox = ({ onSearchHandler, filterString }) => {
42             return <div>
43               <label htmlFor="filter">搜尋</label>
44               <input type="text" id="filter" className="form-control mb-3"
45                 // value={filterString}
46                 // onChange={onSearchHandler}
47                 defaultValue={filterString}
48                 // 換成鍵盤事件Enter後·再發出請求
49                 onKeyDown={onSearchHandler}
50               />
43             </div>
52           }
53
54           // #5a 建立卡片元件
55           const Card = ({ item }) => {
56             return (
57               <div className="card">
58                 <img
59                   src={item.urls.regular}
60                   className="card-img-top img-cover"
61                   alt="..."
62                   height="400"
63                   width="100%"
64                 />
65               </div>
66             )
67           }
68
69           const App = () => {
70             // console.log(1);
71             const [filterString, setFilterString] = useState('cafe');
72             // #3a 建立列表用的陣列
73             const [jsonData, setJsonData] = useState([]);
74
75             // 加入讀取
76             const [isLoading, setIsLoading] = useState(false);
77             // 改用傳參考(變數的相同位址)的方式
78
79             // 使用useRef建立變數
80             const currentPage = useRef(1);
81
82             const onSearchHandler = (e) => {
83               // 查看事件觸發時的相關資訊
84               // console.log(e);
85               // 判斷是否按了Enter
86               if (e.key === 'Enter') {
87                 // 更新資料
88                 setFilterString(e.target.value);
89               }
90             }
91
92             // 把getPhoto轉成獨立函式·並取得特定頁數
93             // 剩餘次數
94             const [count, setCount] = useState(50);
95             const getPhotos = async (page = 1, isNew = true) => {
96               try {
97
98                 // 取得圖片前先寫入
99                 setIsLoading(true);
100
```

```

101 // 加入關鍵字(query)
102
103 const result = await axios.get(`${api}?client_id=${accessKey}&query=${filterString}&page=${page}`);
104 // console.log(result);
105
106 // #3b 更新列表 => 將發出請求的結果更新到列表陣列
107 // setJsonData(result.data.results);
108 setJsonData(preData) => {
109     // 更新資料
110     if (isNew) {
111         return [...result.data.results];
112     }
113     // ...preData => 保存前面出現的內容
114     // ...result.data.result => 目前頁的內容
115     return [...preData, ...result.data.results];
116 });
117
118 // 每次發出請求後，取得剩餘次數 -> 更新
119 setCount(result.headers['x-ratelimit-remaining']);
120
121 // 更新頁數
122 currentPage.current = page;
123
124 } catch (error) {
125     console.log(error);
126 }
127 }
128
129 // #1 建立非同步方法時，會因為渲染的關係不斷重複觸發
130 // 此方法會造成限定次數很快就被用完了
131
132 // const getPhotos = async () => {
133 //     // 手寫程式測試是否能成功連上api
134 //     // console.log(`${api}?client_id=${accessKey}&query=food`);
135
136 //     const result = await axios.get(`${api}?client_id=${accessKey}&query=${filterString}`);
137 //     console.log(result);
138 // }
139 // getPhotos();
140
141 // #2 useEffect
142 // 建立一個列表高度，預設為空值，滑鼠滾動時會產生變化
143 const listRef = useRef(null);
144
145 useEffect(() => {
146     getPhotos(1, true);
147
148     // 建立滾動監聽函式
149     const ScrollEvent = () => {
150         // 垂直捲動位置
151         // console.log('scroll', window.scrollY);
152         // console.dir(listRef.current);
153         // offsetHeight => margin+padding+border
154         // innerHeight => margin+padding
155         // 取得每頁的最後一張圖接近的位置
156         const height = (listRef.current.offsetHeight + listRef.current.offsetTop) - window.innerHeight;
157
158         // 假如（沒有載入圖片）並且（垂直捲軸位置超過目前列表圖片的高度）
159         // 則執行下一頁
160         if (!isLoading.current && window.scrollY > height) {
161             // console.log('換頁了');
162             // 因為換頁了，所以最新頁數要更新
163             currentPage.current = currentPage.current + 1;
164             // 因為只是更新頁數，所以呼叫圖片列表時
165             // 同一個關鍵字的資料不用覆蓋，因此補上false
166             getPhotos(currentPage.current, false);
167         }
168     }
169
170     // 檢測滾動監聽
171     window.addEventListener('scroll', ScrollEvent);
172     // 移除監聽（確保每次捲動時，位置與資料是正確的）
173
174     return () => {
175         window.removeEventListener('scroll', ScrollEvent);
176     }
177 }, [filterString]);
178 // useEffect =>陣列條件改變時，被執行
179 // useEffect =>空陣列時，只會在初使化被執行
180
181 return (<div>
182     <SearchBox onSearchHandler={onSearchHandler} filterString={filterString} />
183     <p>剩餘請求次數:{count}</p>
184
185     /*#4 將陣列資料渲染到畫面上*/
186     /* 使用bootstrap建立一列二欄的格線系統*/
187     <div className="row row-cols-2 g-3" ref={listRef}>
188         {
189             jsonData.map((item) => {
190                 return (
191                     <div className="col" key={item.id}>
192                         /* #5b 呼叫卡片元件*/
193                         <Card item={item} />
194                     </div>
195                 )
196             })
197         }
198     </div>
199
200 </div>
201 }

```

```
202 |
203 |         ReactDOM
204 |             .createRoot(document.querySelector('#root'))
205 |             .render(<App />)
206 |     </script>
207 | </main>
208 | </div>
209 | </div>
210 | <style>
211 |     .img-cover {
212 |         object-fit: cover;
213 |     }
214 | </style>
215 | <link rel="stylesheet" href="./stylesheets/all.css">
216 | <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
217 | </body>
218 |
219 | </html>
```