

#自主學習 由於我的 API 本身並沒有 name 所以如果把搜尋建立在 name 上面是無法的 所以我就想把網站的搜尋做成資料的筆數 我們可以在搜尋欄上面輸入想要搜尋的數量(1-10) 網站便會自動抓取 API

The screenshot shows the homepage of the 'Animals API Project'. It features a search bar at the top labeled 'Search Animals'. Below it are six cards, each containing a small image and the name of an animal:

- BLACK-AND-WHITE COLOBUS MONKEY**
- BURMESE BROWN MOUNTAIN TORTOISE**
- BAT-EARED FOX**
- LAUGHING KOOKABURRA**
- AZUREUS CICHLID**
- MATAMATA**

To the right of the cards, a developer's browser interface is visible, showing the 'Console' tab and a list of animals from the API response. The console output includes:

```
animlas
displayAnimals_xx.js:4
Array(10)
0: {name: 'Black-and-White Colobus Monkey'}
1: {name: 'Burmese Brown Mountain Tortoise'}
2: {name: 'Bat-eared Fox', latin_name: 'Otocyon megalotis'}
3: {name: 'Laughing Kookaburra', latin_name: 'Dacelo novaeguineae'}
4: {name: 'Azureus Cichlid', latin_name: 'Cichlasoma azurinum'}
5: {name: 'Matamata', latin_name: 'Chelus fimbriatus'}
6: {name: 'Trumpeter Swan', latin_name: 'Cygnus buccinator'}
7: {name: 'Sumatran Orangutan', latin_name: 'Pongo abelii'}
8: {name: 'Rhinoceros Hornbill', latin_name: 'Buceros rhinoceros'}
9: {name: 'Cape Porcupine', latin_name: 'Hystrix africaeaustralis'}
length: 10
[[Prototype]]: Array(0)
```

搜尋要的筆數後 就可以抓取資料

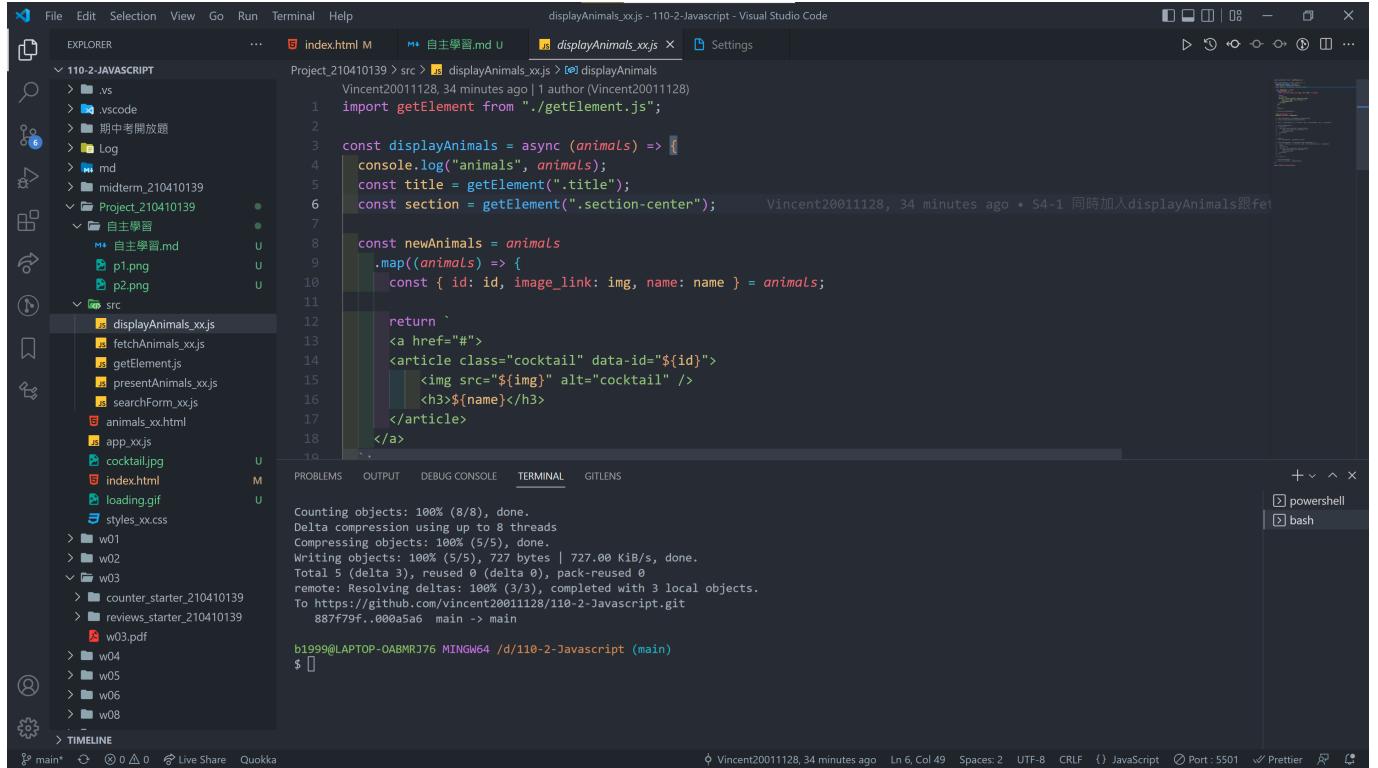
The screenshot shows the homepage after a search for 10 results. The cards displayed are:

- MATAMATA**
- RINGTAIL**
- BURMESE BROWN MOUNTAIN TORTOISE**
- RED AND YELLOW BARBET**
- RED-CROWNED CRANE**
- LESSER KUDU**

The developer's browser interface on the right shows the expanded list of animals from the API response. The console output includes:

```
animlas
displayAnimals_xx.js:4
(10) [{}]
0: {name: 'Matamata', latin_name: 'Chelus fimbriatus'}
1: {name: 'Ringtail', latin_name: 'Bassaris astutus'}
2: {name: 'Burmese Brown Mountain Tortoise', latin_name: 'Cuora amboinensis'}
3: {name: 'Red and Yellow Barbet', latin_name: 'Capitoniger'}
4: {name: 'Red-crowned Crane', latin_name: 'Grus japonensis'}
5: {name: 'Lesser Kudu', latin_name: 'Tragelaphus imberbis'}
6: {name: 'Kangaroo', latin_name: 'Macropus'}
7: {name: 'Hippopotamus', latin_name: 'Hippopotamus'}
8: {name: 'Emerald Tree Boa', latin_name: 'Corallus caninus'}
length: 9
[[Prototype]]: Array(0)
input
searchForm_xx.js:12
animlas
displayAnimals_xx.js:4
(2) [{}]
0: {name: 'Red-capped Cardinal', latin_name: 'Cardinalis sinuatus'}
1: {name: 'Fennec Fox', latin_name: 'Vulpes zerda'}
length: 2
[[Prototype]]: Array(0)
input
searchForm_xx.js:12
input 6
searchForm_xx.js:12
animlas
displayAnimals_xx.js:4
0: {name: 'Fairy Bluebird', latin_name: 'Sialia currucoides'}
1: {name: 'Poison Dart Frog', latin_name: 'Dendrobates tinctorius'}
2: {name: 'Matamata', latin_name: 'Chelus fimbriatus'}
3: {name: 'Barn Owl', latin_name: 'Tyto alba'}
4: {name: 'Tawny Frogmouth', latin_name: 'Podargus strigoides'}
5: {name: 'Bearcat', latin_name: 'Arctictis binturong'}
length: 6
[[Prototype]]: Array(0)
input
searchForm_xx.js:12
input 9
searchForm_xx.js:12
animlas
displayAnimals_xx.js:4
(9) [{}]
0: {name: 'Matamata', latin_name: 'Chelus fimbriatus'}
1: {name: 'Ringtail', latin_name: 'Bassaris astutus'}
2: {name: 'Burmese Brown Mountain Tortoise', latin_name: 'Cuora amboinensis'}
3: {name: 'Red and Yellow Barbet', latin_name: 'Capitoniger'}
4: {name: 'Red-crowned Crane', latin_name: 'Grus japonensis'}
5: {name: 'Lesser Kudu', latin_name: 'Tragelaphus imberbis'}
6: {name: 'Kangaroo', latin_name: 'Macropus'}
7: {name: 'Hippopotamus', latin_name: 'Hippopotamus'}
8: {name: 'Emerald Tree Boa', latin_name: 'Corallus caninus'}
length: 9
[[Prototype]]: Array(0)
input
searchForm_xx.js:12
```

#解決問題 當初在抓資料的時候一直出現錯誤 後來看一看發現第一行紅色的 getelement 是 getelement 第 5、6 行卻是 get 所以導致抓不到 後來 5、6 行改成 getelement 就成功了



```
import getElement from "./getElement.js";

const displayAnimals = async (animals) => [
  console.log("animals", animals);
  const title = getElement(".title");
  const section = getElement(".section-center");

  const newAnimals = animals
    .map((animal) => {
      const { id, image_link, name } = animal;

      return `
        <a href="#">
          <article class="cocktail" data-id="${id}">
            
            <h3>${name}</h3>
          </article>
        </a>
      `;
    })
    .join("");

  return newAnimals;
}

displayAnimals([{"id": "w01", "name": "Wombat", "image_link": "w01.png"}, {"id": "w02", "name": "Kangaroo", "image_link": "w02.png"}, {"id": "w03", "name": "Koala", "image_link": "w03.pdf"}, {"id": "w04", "name": "Emu", "image_link": "w04.png"}, {"id": "w05", "name": "Tasmanian Devil", "image_link": "w05.png"}, {"id": "w06", "name": "Platypus", "image_link": "w06.png"}, {"id": "w07", "name": "Kangaroo", "image_link": "w07.png"}, {"id": "w08", "name": "Kangaroo", "image_link": "w08.png"}]);
```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a project structure for "110-2-JAVASCRIPT". The "src" folder contains files like "displayAnimals\_xx.js", "fetchAnimals\_xx.js", "getElement.js", "presentAnimals\_xx.js", "searchForm\_xx.js", "animals\_xx.html", "app\_xx.js", "cocktail.jpg", "index.html", "loading.gif", and "styles\_xx.css". It also lists several "wXX" files.
- Terminal:** Shows a git commit message and a command prompt:

```
Counting objects: 100% (8/8), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 727 bytes | 727.00 KiB/s, done.  
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.  
To https://github.com/vincent20011128/110-2-Javascript.git  
 887f79f..000ea5a6 main -> main
```

```
b1999@LAPTOP-OABMRJ76 MINGW64 /d/110-2-Javascript (main)
```
- Status Bar:** Shows the current user (Vincent20011128), date (34 minutes ago), line (Ln 6), column (Col 49), spaces (Spaces: 2), encoding (UTF-8), CRLF, file type (JavaScript), port (Port: 5501), and Prettier status.