

Github URL

W14-P1: Get Cocktail data by first letter

The screenshot displays a web application running in a browser. The left pane shows the source code for `index.html`. The code defines a button click event listener that fetches cocktail data from `https://www.thecocktaildb.com/api/json/v1/1/search.php?f=a`. The fetched data is then mapped to a list of cocktail names starting with 'A' and displayed in the browser. The right pane shows the browser's developer tools, specifically the 'AJAX' tab, which displays the JSON response from the API. The response is an array of cocktail objects, each containing details like `idDrink`, `strDrink`, `strDrinkAlternate`, `strTags`, and `strVideo`. The cocktails listed include A1, ABC, Ace, ACID, Adam, AT&T, A. J., Avalon, Apello, Affair, Abilene, Almeria, Addison, Applear, Acapulco, Affinity, Aviation, After sex, and Applejack.

```
const btn = document.querySelector(".btn");

btn.addEventListener("click", async () => {
  const url = "https://www.thecocktaildb.com/api/json/v1/1/search.php?f=a";
  const options = {
    // method: GET,
    // headers: {
    //   "X-RapidAPI-Key": "SIGN-UP-FOR-KEY",
    //   "X-RapidAPI-Host": "the-cocktail-db.p.rapidapi.com",
    // },
  };
  try {
    const response = await fetch(url);
    const data = await response.json();
    console.log(data.drinks);
    displayItems(data.drinks);
  } catch (error) {
    console.error(error);
  }
});

const displayItems = (items) => {
  const displayData = items
    .map((item) => {
      const { strDrink } = item;
      return `<p>${strDrink}</p>`;
    })
    .join("");
  const element = document.createElement("div");
  element.innerHTML = displayData;
}
```

AJAX

show json

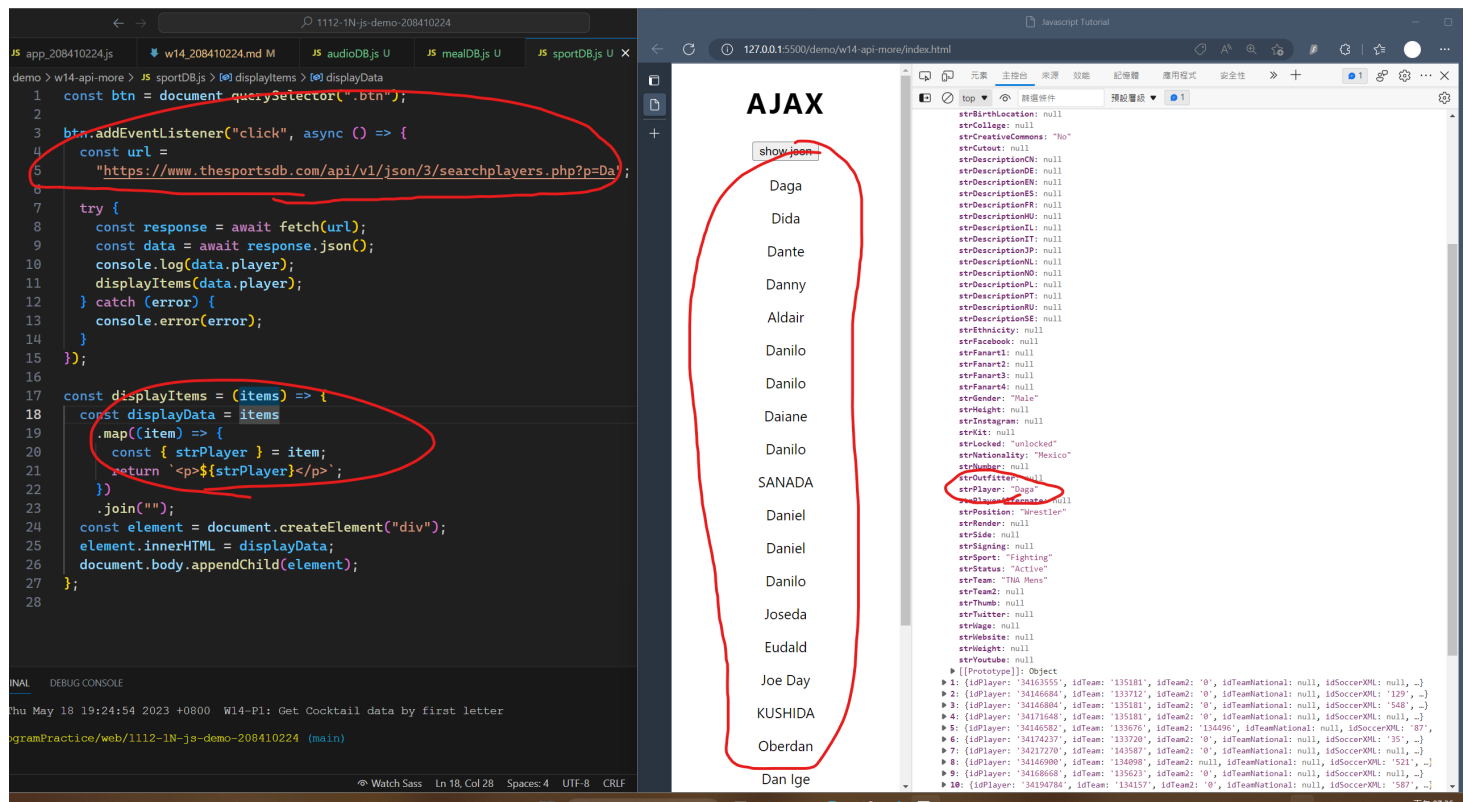
A1
ABC
Ace
ACID
Adam
AT&T
A. J.
Avalon
Apello
Affair
Abilene
Almeria
Addison
Applear
Acapulco
Affinity
Aviation
After sex
Applejack

250ff0c changiojen1

Thu May 18 19:24:54 2023 +0800

W14-P1: Get Cocktail data by first letter

MealDB



AudioDB

The screenshot shows a web application with a button labeled "show json". Clicking the button triggers an AJAX call to the AudioDB API. The code on the left is as follows:

```
1 const btn = document.querySelector(".btn");
2
3 btn.addEventListener("click", async () => {
4   const url =
5     "https://theaudiodb.com/api/v1/json/2/discography.php?s=ed%20sheeran";
6
7   try {
8     const response = await fetch(url);
9     const data = await response.json();
10    console.log(data.album);
11    displayItems(data.album);
12  } catch (error) {
13    console.error(error);
14  }
15 });
16
17 const displayItems = (items) => {
18   const displayData = items
19     .map((item) => {
20       const { strAlbum } = item;
21       return `<p>${strAlbum}</p>`;
22     })
23     .join("");
24   const element = document.createElement("div");
25   element.innerHTML = displayData;
26   document.body.appendChild(element);
27 };
28
```

The browser on the right shows the resulting HTML output, which is a list of albums by Ed Sheeran:

- Eyes Closed
- Celestial
- 2step (The Remixes)
- Merry Christmas
- Bad Habits
- =
- BLOW
- South of the Border
- No.6 Collaborations Project

The browser's developer tools also show the JSON data returned by the API:

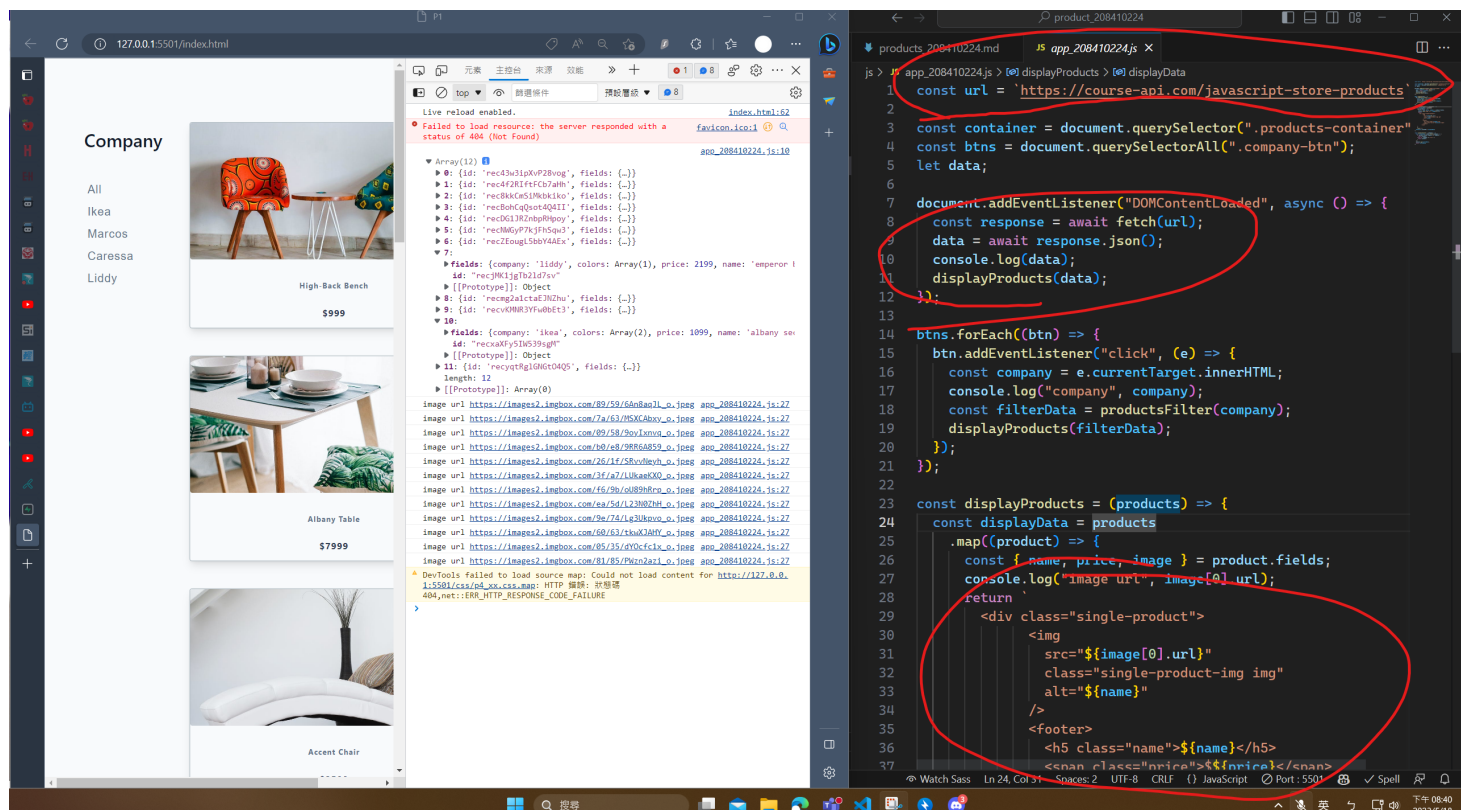
```
[{"strAlbum": "Eyes Closed", "intYearReleased": "2023"}, {"strAlbum": "Celestial", "intYearReleased": "2022"}, {"strAlbum": "2step (The Remixes)", "intYearReleased": "2022"}, {"strAlbum": "Merry Christmas", "intYearReleased": "2021"}, {"strAlbum": "Bad Habits", "intYearReleased": "2021"}, {"strAlbum": "=", "intYearReleased": "2021"}, {"strAlbum": "BLOW", "intYearReleased": "2019"}, {"strAlbum": "South of the Border", "intYearReleased": "2019"}, {"strAlbum": "No.6 Collaborations Project", "intYearReleased": "2019"}]
```

04f81d8 changiojen1

Thu May 18 19:51:34 2023 +0800 W14-P2: Get data from MealDB, SportDB, AudioDB one by one

W14-P3: Get products from url (<https://course-api.com/javascript-store-products>)

Vercel URL



Btn filter

The screenshot displays a web application with a 'Company' filter dropdown on the left. The dropdown menu is open, showing options: 'All', 'Ikea', 'Marcos', 'Caressa', and 'Liddy'. The main content area shows three product cards: 'Accent Chair' (\$2599), 'Wooden Table' (\$4599), and 'Dining Table' (\$699). The background is dark, and the text is white. The interface is annotated with red circles highlighting the filter buttons and the product list.

The code on the right shows the JavaScript logic for the filter. The `btns.forEach()` loop iterates over the filter buttons, and the `btn.addEventListener()` function handles the click event. The `productsFilter` function filters the products based on the selected company. The `displayProducts` function updates the product list.

```
const btns = document.querySelectorAll(".company-btn");
let data;

document.addEventListener("DOMContentLoaded", async () => {
  const response = await fetch(url);
  data = await response.json();
  console.log(data);
  displayProducts(data);
});

btns.forEach(btn => {
  btn.addEventListener("click", (e) => {
    const company = e.currentTarget.innerHTML;
    const filterData = productsFilter(company);
    displayProducts(filterData);
  });
});

const displayProducts = (products) => {
  // ...
};

const productsFilter = (company) => {
  const filterData = data.filter((product) => {
    if (product.fields.company === company) {
      console.log("company", company);
      return product;
    } else if (company === "all") {
      console.log("company all");
      return product;
    }
  });
  return filterData;
};
```