

Laboratorio Módulo 9 - Llamadas asíncronas

Acabamos de recibir un encargo de *Netflix*. Quieren promocionar una de sus conocidas series, *Breaking Bad*, y necesitan una pequeña aplicación creada con *Javascript* para mostrar información sobre sus personajes.

La aplicación debe mostrar un listado con la foto y nombre de los personajes (*characters*) de la serie. Al hacer click sobre uno de ellos se mostrará más información acerca del mismo: foto, nombre, fecha de nacimiento y apodo.

Según nos comentan, podemos recuperar los datos de su API. Nos envían esta documentación: [The Breaking Bad API](#)

En esta documentación hemos podido ver que la ruta base de la api es "https://www.breakingbadapi.com/api/", y tenemos disponibles los siguientes endpoints:

- Recuperar todos los personajes: `/api/characters`
- Recuperar un personaje: `/api/characters/{char_id}`

Para crear el listado y el detalle, nos facilitan un fichero *utils.js* del que nos detallan que debemos utilizar la siguiente funcionalidad:

- **createCharacter**: debemos utilizar esta función para que nos devuelva un elemento HTML con la fila del personaje que recibe por parámetro.
- **showCharacter**: debemos utilizarla para que nos muestre el detalle del personaje que recibe por parámetro.

El contenido del fichero *utils.js* es el siguiente:

```
const createCharacterRow = character => {
  const element = document.createElement("div");

  const avatar = createAvatar(character);
  element.appendChild(avatar);

  const link = createRowText(character);
  element.appendChild(link);

  element.className = "character-row";

  return element;
};

const createAvatar = character => {
  const element = document.createElement("img");
  element.width = 150;
  element.className = "thumbnail";
  element.src = character.img;

  return element;
};
```

```
};

const createRowText = character => {
  const element = document.createElement("span");
  element.append(character.name);

  return element;
};

const createAvatarDetail = character => {
  const element = document.createElement("img");
  element.width = 350;
  element.src = character.img;

  return element;
};

const showCharacter = character => {
  console.log("character", character);
  const characterDetail = document.getElementById("character-detail");

  characterDetail.innerHTML = "";
  characterDetail.appendChild(createAvatarDetail(character));
  characterDetail.appendChild(createParagraph("Name: " + character.name));
  characterDetail.appendChild(
    createParagraph("Birthday: " + character.birthday)
  );
  characterDetail.appendChild(
    createParagraph("Nickname: " + character.nickname)
  );
};

const createParagraph = text => {
  const element = document.createElement("p");
  element.append(text);
  return element;
};

export { createCharacterRow, showCharacter };
```

Además sus diseñadores nos han facilitado el siguiente fichero `src/styles.css` para darle un poco de estilo a la primera versión de prueba que le entregaremos de la aplicación:

```
body {
  font-family: sans-serif;
  display: flex;
  flex-direction: column;
  background-color: #eee;
}

.character-row {
```

```
    cursor: pointer;
  }

#character-detail {
  position: fixed;
  right: 0;
  padding: 25px;
  background-color: #666;
  color: white;
  font-size: 140%;
  height: 100vh;
}

.thumbnail {
  border-radius: 50%;
  margin: 15px;
  vertical-align: middle;
}
```

y un fichero *index.html*:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
  </head>
  <body>
    <h1>Breaking Bad Characters</h1>
    <div id="root">Loading...</div>
    <div id="character-detail"></div>
    <script src="./src/index.js"></script>
  </body>
</html>
```

Para importar el fichero css sólo tendremos que añadir al principio de nuestro *src/index.js* la siguiente línea:

```
import "./styles.css";
```

Nos comentan los siguientes puntos:

- Separar un fichero *data-business.js* con la funcionalidad de comunicación con su API.
- Podemos utilizar *fetch* o *axios* para realizar las peticiones de datos.
- Podemos entregar la aplicación en un *.zip* o hacer un fork de un *CodeSandbox* inicial que nos pasan con el contenido de los ficheros anteriores: [breaking-bad-characters - CodeSandbox](#)

Además:

- De forma **obligatoria** para que puedan evaluar nuestro trabajo, tendremos que entregar la aplicación con el listado de personajes.
- Como **extra**, tendremos que hacer funcionar el click en cada personaje para mostrar el detalle del mismo.
- **¡Y nos proponen un par de retos!**: -- Utilizar otro de los endpoints de su API para recuperar y mostrar más datos, como un listado de episodios o frases célebres de cada personaje para mostrar en su detalle. -- Hacer otra aplicación similar, pero utilizando la API de la serie **Rick y Morty**. Aquí nos dejan su documentación: [The Rick and Morty API](#)