




Blog with API

Marc Parellada & Adrián Sánchez



What have we learnt with this project.

Bootstrap 5

We have learnt how to build a website with Bootstrap, by implementing its layout and container system that speed up the development of a responsive web, and by using complements such as *collapsible* blocks, *modals* and *accordions*.



What have we learnt with this project.

JSON Server

Here at the school we had troubles to fetch data from <https://jsonplaceholder.typicode.com/> because the IP address was blocked. In order to use their API data, we had to install JSON Server module with NPM. Then, we had to run `npx json-server --watch <db.json location>` in order to have a server that responds to our request operations.

By using this options, we could update and delete data from the website and request the changes to the API.

A blue rectangular box with a subtle pattern of small white dots. The text "JSON-Server" is centered in white, with "JSON" in a larger font size than "Server".

JSON-Server

What have we learnt with this project.

Fetch API & Async/Await

With the Fetch API we could make request asynchronously across the network in a simpler way. This method returns a promise, that once has been fulfilled, returns a response whose methods can return another promise.

To save the requested data to use it later, instead of using it inside a promise callback, we used the *await* to save the result of a promise (which is an asynchronous operation) in a variable or pause the execution until has finished. This allowed us to create *request functions* which are independent to the treatment of the data, and write code in the same way we would do it as it was synchronous.

```
export async function getUser(id) {
  const url = `${sessionStorage.url}/users/${id}`;
  const response = await fetch(url);
  const user = await response.json();

  return user;
}

export async function updatePost(id, data) {
  const url = `${sessionStorage.url}/posts/${id}`;
  const options = {
    method: "PATCH",
    body: JSON.stringify(data),
    headers: {
      "Content-Type": "application/json; charset=utf-8",
    },
  };

  const response = await fetch(url, options);

  return response;
}

export async function deletePost(id) {
  const url = `${sessionStorage.url}/posts/${id}`;
  const options = {
    method: "DELETE",
  };

  const response = await fetch(url, options);

  return response;
}
```

Problems

Bootstrap can speed up the development...

But it forces you to know its rules and stick to them. We had to understand and study how its classes alter functionality of its components. Some components use its own *JavaScript* methods to interact with them. For example, *modals* and *collapse* components have built-in toggle, show and hide methods and events that we haven't finished mastering.

We were not able to have more than one modal, so we had to make collapsable sections to display the post content, the post editor and the remove confirmation message.

Our approach wasn't to apply many custom styles, so it has not been a problem. But if it had been it, we would have used SASS, which is actually out of our scope.

Problems

Page reloading

When we requested changes such as post updates or deletes, once the change was reflected in *db.json* file the page was reloaded. This was caused by the *Live Server* extension every time it detected a change in a file in the working directory. *Live server* watches out for saving changes, and consequently it updates.

To fix this issue, we had to move the *data* folder outside the working area and run the *json-server* with a different path for *db.json*.

Problems

Pagination

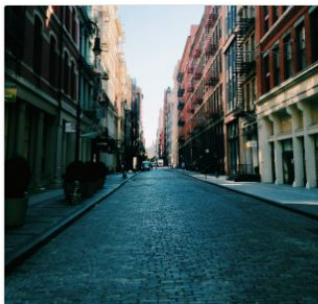
It takes up to 3 hours to complete. it wasn't supposed to be too complicated, but the first approach was to display 3 numeric boxes for page selection, and 2 more boxes to go to the previous and next page. At the end, we only left 1 numeric box that indicates the current page.



qui et at rerum necessitatibus

aut est omnis dolores neque rerum quod ea rerum velit pariatur beatae excepturi et provident voluptas corrupti corporis harum reprehenderit dolores...

Author: Mrs. Dennis Schulist, Karley_Dach@jasper.info



sed ab est est

at pariatur consequuntur earum quidem quo est laudantium soluta voluptatem qui ullam et est et cum voluptas voluptatum repellat est

Author: Mrs. Dennis Schulist, Karley_Dach@jasper.info



voluptatum itaque dolores nisi et quasi

veniam voluptatum quae adipisci id et id quia eos ad et dolorem aliquam quo nisi sunt eos impedit error ad similique veniam

Author: Mrs. Dennis Schulist, Karley_Dach@jasper.info



And that's all guys

