**npm**

Let's get back to the topic of fetching data from the server.

We could use the previously mentioned promise based function [fetch](https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/fetch) to pull the data from the server. Fetch is a great tool. It is standardized and supported by all modern browsers (excluding IE).

That being said, we will be using the [axios](https://github.com/axios/axios" \t "_blank) library instead for communication between the browser and server. It functions like fetch, but is somewhat more pleasant to use. Another good reason to use axios is our getting familiar with adding external libraries, so-called *npm packages*, to React projects.

Nowadays, practically all JavaScript projects are defined using the node package manager, aka [npm](https://docs.npmjs.com/getting-started/what-is-npm" \t "_blank). The projects created using create-react-app also follow the npm format. A clear indicator that a project uses npm is the *package.json* file located at the root of the project:

Texto

Descripción generada automáticamente

At this point the *dependencies* part is of most interest to us as it defines what *dependencies*, or external libraries, the project has.

We now want to use axios. Theoretically, we could define the library directly in the *package.json* file, but it is better to install it from the command line.



**NB npm-commands should always be run in the project root directory**, which is where the *package.json* file can be found.

Axios is now included among the other dependencies:

Texto

Descripción generada automáticamente

In addition to adding axios to the dependencies, the npm install command also *downloaded* the library code. As with other dependencies, the code can be found in the *node\_modules* directory located in the root. As one might have noticed, *node\_modules* contains a fair amount of interesting stuff.

Let's make another addition. Install *json-server* as a development dependency (only used during development) by executing the command:



and making a small addition to the *scripts* part of the *package.json* file:

Texto

Descripción generada automáticamente

“server”: “json-server -p3001 –watch db.json”

We can now conveniently, without parameter definitions, start the json-server from the project root directory with the command:

