

# AJAX & fetch

## Empezando con fetch en Javascript

Fetch es una interfaz para hacer solicitudes **AJAX** en **JavaScript**. Actualmente es ampliamente implementado por los navegadores modernos y es usado generalmente para hacer una solicitud a un API.

```
const promesa = fetch(url, [options]);
```

Una solicitud **fetch** regresa una promesa, la cual es un objeto del tipo **Promesa (promise)**, que finalmente resolverá con un objeto del tipo **Response**. La promesa será rechazada si sucede un error de red, y será resuelta si no hay problema conectando al servidor y el servidor responde con algún código de estatus. Este estatus podría ser 200's, 400's o 500's.

```
fetch(url)
  .then(response ⇒ response.json())
  .catch(err ⇒ console.log(err))
```

La solicitud es enviada con el **método GET** por defecto. Para enviar una solicitud del tipo POST / PATCH / DELETE / PUT, se debe utilizar la propiedad **method** como parte del parámetro **options**. Algunos otros valores posibles para **options** son:

- **method**: tales como GET, POST, PATCH
- **headers**: un objeto del tipo Header
- **mode**: tales como cors, no-cors, same-origin
- **cache**: el modo de cache para la solicitud
- **credentials**: en ocasiones requeridas para identificarse en la API destino
- **body**: el contenido de la solicitud

El siguiente ejemplo muestra el uso de **fetch**, la cual hace una solicitud a la API de GitHub para obtener una lista de repositorios git.

```
const url = 'https://api.github.com/users/shrutikapoor08/repos';

fetch(url)
  .then(response ⇒ response.json())
  .then(repos ⇒ {
    const reposList = repos.map(repo ⇒ repo.name);
    console.log(reposList);
  })
  .catch(err ⇒ console.log(err))
```

Para hacer una solicitud con el **método POST**, utilizaremos el parámetro **method**. En el siguiente ejemplo se hace una solicitud asíncrona, con la sintaxis **async/await**.

```
const params = { id: 123 }

const response = await fetch('url', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify(params)
});

const data = await response.json();
```