



DigitalHouse>

Fetch y Axios



**Certified Tech
Developer**

The Ultimate Degree

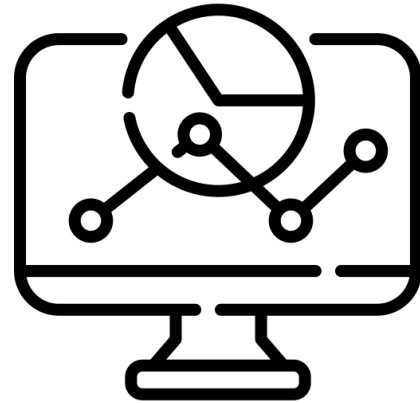
Índice

1. [Fetch](#)
2. [Axios](#)

1 | Fetch

¿Qué es Fetch?

Fetch es una API JavaScript nativa para navegadores (no hay que instalar nada) que permite realizar peticiones HTTP asíncronicas por medio de promises. Fetch simplifica enfoques previos basados en objetos XMLHttpRequest con un código más legible y práctico.



Descripción

La sintaxis de Fetch es la siguiente:

```
fetch("/noBombardeenLasToninas.api").then(function(respuesta) { ... });
```

De este modo, retorna una promesa que solo es rechazada si ocurre un fallo de red o una abuela georgiana se cruza en el camino.

```
fetch(url)
  .then((res) => res.json())
  .then((res) => console.log(res));
```

El primer método `then()` recibe un string por lo que se le aplica el método nativo `json()` para convertirlo en un objeto legible por JavaScript. En el segundo `then()` usaremos la lógica que deseemos con los datos obtenidos.

Options

La función Fetch tiene un segundo parámetro: un objeto opcional que permite configurar la petición. En este código podemos ver cómo hacer la petición con Fetch:

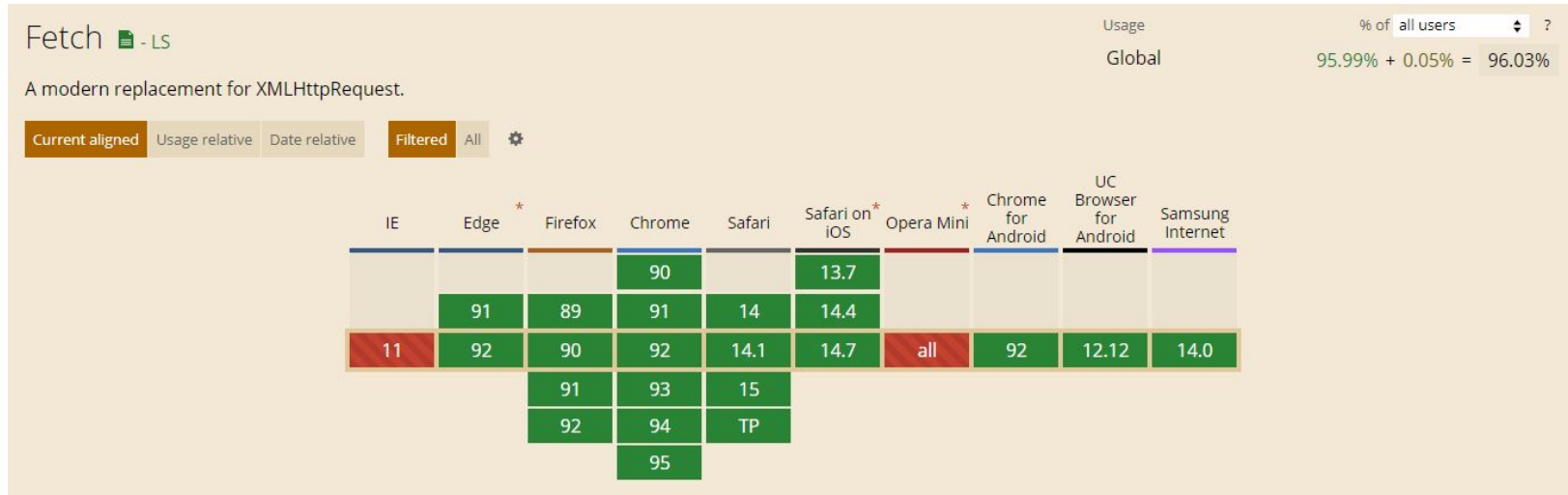
```
const options = {  
  method: "POST",  
  credentials: "include",  
  headers: { "Content-Type": "application/json" },  
  body: datos ? JSON.stringify(datos) : null,  
}  
  
fetch(url, options);
```

Tabla comparativa

JSON es el acrónimo de JavaScript Object Notation y, como su nombre lo indica, es muy similar al objeto literal que ya conocemos. Veamos las diferencias:

Options	
method	Método HTTP. Por ejemplo: POST, GET, PUT, DELETE y HEAD.
credentials	Permite cambiar el modo en el que se realiza el request. Por ejemplo, habilitar el envío de cookies al servidor.
headers	Son las cabeceras HTTP que permiten –tanto al cliente como al servidor– enviar información extra. Existen un gran número de opciones. Podés acceder a más info en: https://developer.mozilla.org/es/docs/Web/HTTP/Headers
body	Cuerpo de la petición HTTP. Datos que enviamos al servidor.

Compatibilidad en navegadores



2 | Axios

Axios

Axios es una biblioteca third party (hay que instalarla). Es un cliente HTTP que admite la API Promise y que facilita la configuración y ejecución —tanto desde el lado del cliente como del lado del servidor— a través de un cliente HTTP basado en promesas para Node.js. En el servidor utiliza el módulo HTTP de Node.js y en el navegador usa XMLHttpRequest.



Beneficios

Las principales ventajas de Axios son:

- Interceptar requests y responses.
- Cancelar solicitudes.
- Protección contra XSRF.
- Transformar datos de requests y responses.
- Transformar automáticamente datos JSON.

¿Cómo instalarlo?

Si utilizamos npm, debemos ingresar el comando:

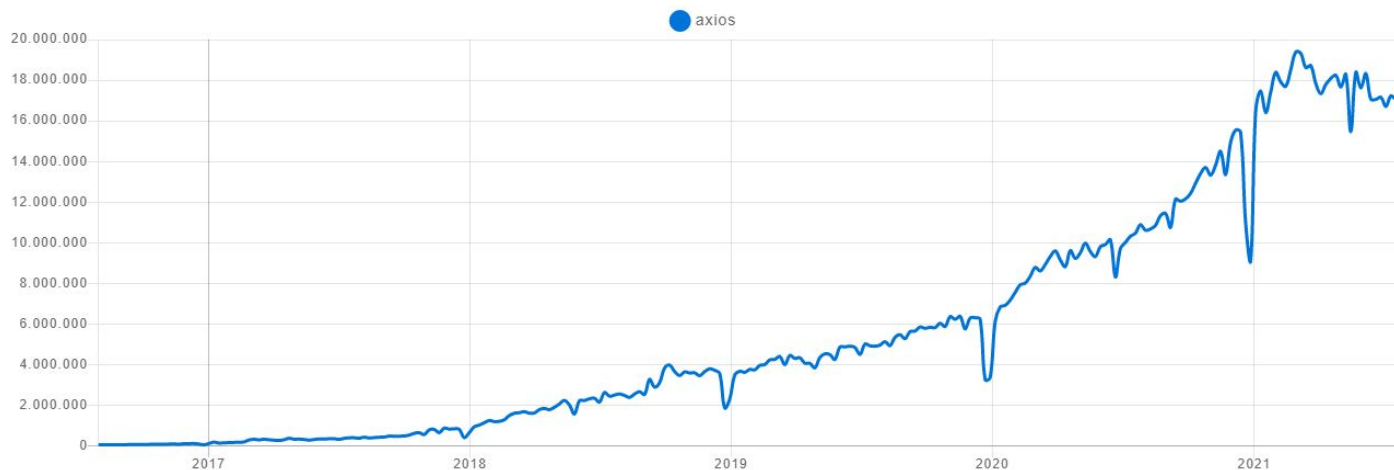
npm install axios

En caso de utilizar Yarn, debemos ingresar el comando:
















yarn add axios

Popularidad

Su nivel de uso por la comunidad JavaScript es alucinante. En la actualidad, ronda las 18.000.000 de descargas semanales en npm, según el sitio <https://www.npmtrends.com/>. ¡Axios mata Bitcoin!



Compatibilidad en browsers

					
Latest ✓	Latest ✓	Latest ✓	Latest ✓	Latest ✓	11 ✓
 Firefox	 Chrome	 IE	 Edge	 Safari	
87  7 ✕	89  7 ✕	11  8.1 ✕	89  10 ✕	9 ✕ 10.11 ✕	
				10 ✕ 10.12 ✕	
				11 ✕ 10.13 ✕	

Diferencias entre Fetch y Axios

fetch	Axios
Es una API nativa. No se instala.	Es un paquete de terceros. Se instala.
Sencillo de usar.	Es aún más sencillo de usar.
No funciona en IE 11.	Funciona en IE 11.
Hay que convertir los datos recibidos a JSON.	Conversión automática.
No usa XMLHttpRequest.	Usa XMLHttpRequest.
Funciona en Node.js si se instala la dependencia node-fetch.	Funciona en Node.js.



Realmente no hay grandes diferencias.
Dependerá de gustos personales o
criterios empresariales.

DigitalHouse>