

FETCH

La función `fetch` permite realizar solicitudes HTTP de manera asíncrona, por lo que la utilizaremos (posteriormente se utilizará AJAX) para interactuar con servicios web y recuperar o enviar datos, `fetch` devuelve una `promise` que resuelve a la respuesta de la solicitud. La respuesta es un objeto que contiene información sobre la respuesta HTTP, como el estado, encabezados y el cuerpo, y se analiza en función del caso.

¿Cómo se utiliza?

`fetch(url_a_la_se_realiza_la_solicitud, otros(método HTTP, encabezados ...))`

```
.then(response => {  
    //Manipulación de la respuesta  
})  
  
.then(data => {  
    // Manipulación de los datos  
})  
  
.catch(error => {  
    // Manejo de errores  
});
```

¿Cómo trabajar con JSON desde Fetch?

Cuando se realiza una solicitud HTTP y la respuesta es json, se utilizará el método `json()` sobre la respuesta para parsear el cuerpo de la respuesta como JSON. Después se manipulan los datos con `data`.

```
fetch('url')  
  
.then(response => response.json())  
  
.then(data => console.log(data))  
  
.catch(error => console.error('Error:', error));
```

GET y POST

Solicitudes GET y POST son métodos HTTP utilizados para enviar información entre el cliente y el servidor.

1. GET: Se utiliza para solicitar datos del servidor, es indicado para operaciones seguras y que no alteran el estado del servidor, puesto que los datos se envían a través de la URL como parámetros de consulta. Los datos son visibles en la barra de direcciones del navegador y en los registros del servidor.

```
fetch('http://localhost:3000/api/tareas?nombre=Ejemplo&edad=25')  
  
.then(response => response.json())  
  
.then(data => console.log(data))  
  
.catch(error => console.error('Error:', error));
```

2. POST: Se utiliza para enviar datos al servidor para ser procesados y modificar el estado del servidor, este es indicado para operaciones no seguras que pueden alterar el estado del servidor. Los datos se envían en el cuerpo (body) de la solicitud, por lo que no se ven directamente en la URL. Los datos no son visibles directamente en la barra de direcciones y no quedan registrados en los registros del servidor.

```
const datosAEnviar = {
```

```
    nombre: "Ejemplo",  
    edad: 25  
  };  
  const opciones = {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json'  
    },  
    body: JSON.stringify(datosAEnviar)  
  };  
  fetch('http://localhost:3000/api/tareas', opciones)  
    .then(response => response.json())  
    .then(data => console.log('Respuesta del servidor:', data))  
    .catch(error => console.error('Error al realizar la solicitud:', error));
```