
SENPAI
ACADEMY

Fetch 📦

SENPAI
ACADEMY

Fetch - Intro

Fetch es el nombre de una nueva API para Javascript con la cual podemos realizar **peticiones HTTP asíncronas** utilizando **promesas.**

La forma de realizar una petición es muy sencilla, básicamente se trata de llamar a **fetch** y pasarle por parámetro la **URL** de la petición a realizar y de forma opcional, un objeto options con opciones de la petición HTTP.

```
// Realizamos la petición y guardamos la promesa
const request = fetch("https://google.com");

// Si es resuelta, entonces ejecuta esta función...
request.then(function(response) { ... });
```

SENPAI ACADEMY

Fetch - Opciones

El parámetro opcional **options** podemos definir varios opciones, los más importantes son:

- **method**: Método HTTP de la petición. Por defecto, GET.
- **body**: Cuerpo de la petición HTTP. Puede ser de varios tipos: String, FormData, Blob, etc.
- **headers**: Cabeceras HTTP.

```
// Opciones de la petición (valores por defecto)
const options = {
  method: "GET"
};

// Realizamos la petición y guardamos la promesa
const request = fetch("https://google.com", options);

// Esperar la respuesta con then
request.then(response => response.text())
        .then(data => {
          /** Procesar los datos **/
        });
```

SENPAI ACADEMY

Fetch - body

En caso de que quisiéramos enviar datos al servidor, debemos utilizar la opción **body**.

```
const options = {
  method: "POST",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify(datosAEnviarObj)
};
```

Fetch - response

Si volvemos a nuestro ejemplo de la petición con **fetch**, observaremos que en el primer **.then()** tenemos un objeto **response**. Se trata de **la respuesta que nos llega del servidor web** al momento de recibir nuestra petición.

```
// Petición HTTP
fetch("https://google.com", options)
  .then(response => response.text())
  .then(data => {
    /** Procesar los datos **/
  });
```

SENPAI ACADEMY

Promises 😔

SENPAI
ACADEMY

Promises - Intro

Las promesas son objetos de JavaScript que nos sirven para representar un valor que puede estar disponible **ahora**, en el **futuro**, o **nunca**.

Cuando una función o método nos **retorna una promesa**, esa promesa se va a encontrar en uno de estos estados:

- **pendiente** (pending): estado inicial
- **cumplida** (fulfilled): significa que la operación se completó con éxito.
- **rechazada** (rejected): significa que la operación falló.

SENPAI ACADEMY

Promises - then y catch

En las promesas contamos con los siguientes **métodos** para **ejecutar código dependiendo del resultado**:

- **.then():** recibe una declaración de una **función** y la misma es ejecutada una vez la request se **completó correctamente**.

- **.catch():** recibe una declaración de una función y la misma es ejecutada si la request da **error**.

SENPAI ACADEMY

Promises - then y catch

```
// Petición HTTP retorna una promise
 fetch('https://senpaiacademy.com/uy')
    .then((response) => {
        console.log(response)
    })
    .catch((response) => {
        console.log('Error!');
    })
```

SENPAI
ACADEMY

LINKS 🔗

SENPAI
ACADEMY

LINKS

- Fetch
  - https://lenguajejs.com/javascript/peticiones-http/fetch/
  - https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch
- Promises
  - https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Promise

SENPAI ACADEMY

ICONOS

SENPAI
ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE