

Programador JavaScript Avanzado

Unidad 5: Pipeline y Fetch

Indice

Unidad 5: Pipeline y Fetch

- API fetch



Objetivos

Que el alumno logre:

- Implementar pipeline y fetch



API fetch

fetch es una API de JavaScript que proporciona una forma moderna y flexible de hacer solicitudes de red, como peticiones HTTP. Permite realizar solicitudes de forma asíncrona, lo que significa que puedes enviar y recibir datos desde un servidor web sin bloquear la ejecución del resto del código. **fetch** está construido sobre promesas, lo que facilita la escritura de código asíncrono más limpio y legible.

La sintaxis básica de **fetch** se ve así:

```
fetch(url)
  .then(response => {
    // Manejar la respuesta del servidor
  })
  .catch(error => {
    // Manejar errores de red u otros errores
  });
```

fetch toma como argumento una URL y devuelve una promesa que se resuelve en un objeto **Response** representando la respuesta a la solicitud. Puedes encadenar **.then()** para manejar la respuesta del servidor y convertirla en el formato que necesitas (por ejemplo, JSON). También puedes encadenar un **.catch()** para manejar errores de red u otros problemas que puedan ocurrir durante la solicitud.

Veamos un ejemplo de cómo usar **fetch** para hacer una solicitud GET y manejar la respuesta como JSON:

```
fetch('https://api.example.com/data')
  .then(response => {
    if (!response.ok) {
      throw new Error('La solicitud no fue exitosa');
    }
    return response.json(); // Convierte la respuesta a JSON
  })
  .then(data => {
    console.log(data); // Manejar los datos recibidos del servidor
  })
  .catch(error => {
    console.error('Error durante la solicitud:', error);
  });
```

fetch también admite diferentes métodos HTTP, como POST, PUT y DELETE, así como la posibilidad de incluir encabezados personalizados en la solicitud. Puedes consultar la documentación para obtener más detalles sobre cómo personalizar las solicitudes según tus necesidades específicas.



Resumen

En esta Unidad...

Trabajamos con Pipeline y fetch

En la próxima Unidad...

Trabajaremos con Paradigmas

