Programador JavaScript Avanzado

Unidad 5: Pipeline y Fetch

Contactos: tecnologia@elearning-total.com
Web: www.elearning-total.com

Indice

Unidad 5: Pipeline y Fetch

Objetivos

Que el alumno logre:

- Implementar pipeline y fetch

API Promise

En JavaScript, una promesa es un objeto que representa el resultado eventual (o el error) de una operación asíncrona. Las promesas son una forma de manejar las operaciones asíncronas de manera más cómoda y legible, evitando el anidamiento excesivo de callbacks (conocido como "callback hell").

Una promesa puede estar en uno de los tres estados:

Pendiente (Pending): La operación asíncrona aún no se ha completado ni ha sido rechazada.

Cumplida (Fulfilled): La operación asíncrona se completó con éxito y la promesa devuelve un valor.

Rechazada (Rejected): La operación asíncrona falló y la promesa devuelve un motivo (generalmente un objeto de error) que indica por qué falló.

Las promesas en JavaScript se crean utilizando el constructor Promise. La sintaxis básica de una promesa es la siguiente:

```
const miPromesa = new Promise((resolve, reject) => {
    // Código asíncrono aquí
    // Si la operación se completa con éxito, llamamos a resolve con el resultado
    // Si la operación falla, llamamos a reject con un motivo (por ejemplo, un objeto de error)
});
```

Puedes encadenar métodos .then() y .catch() para manejar el resultado de una promesa. Por ejemplo:

```
miPromesa .then(resultado => {
    // La operación se completó con éxito, resultado contiene el valor resuelto
    console.log(resultado);
}) .catch(error => {
    // La operación falló, error contiene el motivo del rechazo
    console.error(error);
});
```

Además, las promesas también permiten el uso del método .finally(), que se ejecutará independientemente de si la promesa se cumple o se rechaza.

Las promesas son una herramienta fundamental en la programación asíncrona en JavaScript y son ampliamente utilizadas en el desarrollo web para manejar operaciones como solicitudes de red, lectura de archivos y otras operaciones que llevan tiempo.

Resumen

En esta Unidad…

Trabajamos con Pipeline y fetch

En la próxima Unidad…

Trabajaremos con Paradigmas