

Middleware en una API REST con Node.js y Express

Introducción

En el desarrollo de APIs REST con Node.js y Express, uno de los conceptos más importantes pero muchas veces abstracto para los principiantes es el de los middlewares. En esta guía te explicamos qué son, cómo funcionan, sus diferentes tipos y cómo se relacionan con ejemplos de la vida real.

¿Qué es un Middleware?

¿Qué es un Middleware?

Un middleware es una función que se ejecuta entre la petición del cliente (request) y la respuesta del servidor (response). Puede acceder, modificar o simplemente observar los objetos ``req`` (request), ``res`` (response) y la función ``next``, que permite continuar con la siguiente función en la cadena.

Ejemplo de la vida real:

Imaginemos que una API es como una fábrica. Cuando entra un pedido (request), pasa por varias estaciones (middlewares), donde se hacen verificaciones, modificaciones o registros, antes de llegar al producto final (response). Cada estación decide si deja pasar el pedido o si lo detiene.

Tipos de Middlewares

Tipos de Middlewares

1. ****Middlewares de aplicación****:

Middleware en una API REST con Node.js y Express

Se aplican a toda la aplicación y normalmente se configuran en archivos como `app.js` o `index.js`. Por ejemplo: un logger que registra cada petición que llega al servidor.

```
```js
const express = require('express');

const app = express();

app.use((req, res, next) => {

 console.log(`${req.method} ${req.url}`);

 next();

});
...
```
```

Ejemplo de la vida real:

Como una cámara de seguridad en la puerta principal que registra cada vez que alguien entra o sale del edificio.

2. **Middleware de ruta**:

Se aplican solo a rutas específicas. Suelen usarse para validar datos, verificar permisos o procesar información particular.

```
```js
const validarId = (req, res, next) => {

 if (!Number.isInteger(+req.params.id)) {

 return res.status(400).send("ID inválido");

 }

 next();
}
```

## Middleware en una API REST con Node.js y Express

```
}

 next();
};

app.get('/usuarios/:id', validarId, (req, res) => {
 res.send(`Usuario con ID ${req.params.id}`);
});
...
```

Ejemplo de la vida real:

Como un guardia de seguridad que solo permite el ingreso a ciertas salas a personas autorizadas.

### Conclusión

#### Conclusión

Los middlewares son una herramienta poderosa en Express para manejar tareas repetitivas, seguridad, validaciones y más. Entender cómo y cuándo usarlos puede ayudarte a construir aplicaciones más limpias, organizadas y seguras.