
SENPAI
ACADEMY

Middlewares

SENPAI
ACADEMY

Node.js - Middlewares en Express

Las funcionalidades **middleware** (conocidas también como “Man in the Middle”) son funciones que podemos agregar antes de que se ejecute nuestro manejador para la ruta que se accede. (documentación)

En el **middleware** tenemos acceso a la *request* y al *response*, pero también a la función *next*, que será el siguiente **middleware**, o el manejador de la *request*.

```
app.use('/', function(req, res, next) {
  // Logica del middleware
  // ....
  // Podemos acceder a la request, y agregar datos
  console.log('Metodo de la request: ', req.method);
  req.userName = 'Diego';

  next();
});
```

SENPAI ACADEMY

Node.js - Middlewares en Express

Esto nos sirve para:

- Loguear peticiones
- Realizar cambios en la request o en la response
- Finalizar el ciclo de solicitud/respuesta
- Invocar a la siguiente función del middleware en la lista

Express nos provee un **middleware** para servir contenido estático. Esto es útil para responder requests de CSS, imágenes, pdfs, etc. **Express** va a buscar los archivos relativos a la carpeta estática, por lo que no es necesario poner el nombre de la carpeta en la URL.

```
app.use(express.static(path.join(__dirname, "public")));
```

SENPAI ACADEMY

Node.js - Middlewares en Express

```
// Este middleware se llama con cada request
app.use(function (req, res, next) {
  console.log('Time Entry:', Date.now());
  req.userName = 'Diego';
  next();
});

app.get('/*', function (request, response, next) {
  console.log('UserName en request: ', request.userName);
  response.send("Ruta no encontrada, pruebe nuevamente");
  next();
})

// Este middleware se ejecuta al finalizar la ruta anterior
app.use(function (req, res, next) {
  console.log('Time Exit:', Date.now());
  next();
});

app.listen(4000, function () {
  console.log('El servidor quedó corriendo en el puerto 4000');
});
```

El **middleware** también se puede ejecutar luego de que la ruta se resuelva, pero para esto hay que definir el tercer parámetro en la ruta (función ***next***) e invocar a ***next()*** al finalizar

SENPAI ACADEMY

```
const express = require('express');
const path = require('path');
const app = express();

// Este middleware se llama con cada request
app.use(function (req, res, next) {
  console.log('Time:', Date.now());
  req.userName = 'Diego';
  next();
});

app.use(express.static(path.join(__dirname, "public")));

//Este middleware se llama cuando se invoca cualquier verbo sobre la ruta /user/:id
app.use('/user/:id', function (req, res, next) {
  console.log('Request Type:', req.method);
  next();
});

//Esta ruta se llama cuando se invoca el verbo GET sobre la ruta /user/:id
app.get('/user/:id', function (req, res) {
  res.send('Usuario con Id ' + req.params.id);
});

app.get('/*', function (request, response) {
  console.log('UserName en request: ', request.userName);
  response.send("Ruta no encontrada, pruebe nuevamente");
})

app.listen(4000, function () {
  console.log('El servidor quedó corriendo en el puerto 4000');
});
```

SENPAI
ACADEMY

Express -Librería externas

La funcionalidad de **Express** se puede extender utilizando otras librerías. Estas librerías van a funcionar como middlewares en **Express**.

Algunas de las librerías más comunes para utilizar con Express son:

- body-parser => Cuando recibamos requests con el verbo POST nos va a ayudar a manejar el cuerpo
- CORS => Configurar Cross Domain, desde que IPs o Dominios aceptamos requests

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();

// parse application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended: false }));

// parse application/json
app.use(bodyParser.json());
```

SENPAI ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE