



UNIVERSIDAD DE COLIMA

FACULTAD DE TELEMÁTICA.

INGENIERÍA EN SOFTWARE.

COLIMA, COLIMA, 21 DE MARZO DEL 2022.

ASIGNATURA: PROGRAMACIÓN PARA WEB.

CATEDRÁTICO: ULIBARRI IRETA CARLOS.

PRÁCTICA 9. MIDDLEWARE

ALUMNO: OCAMPO BARAJAS JOSÉ RODOLFO.

4 H.

¿Qué es un middleware?

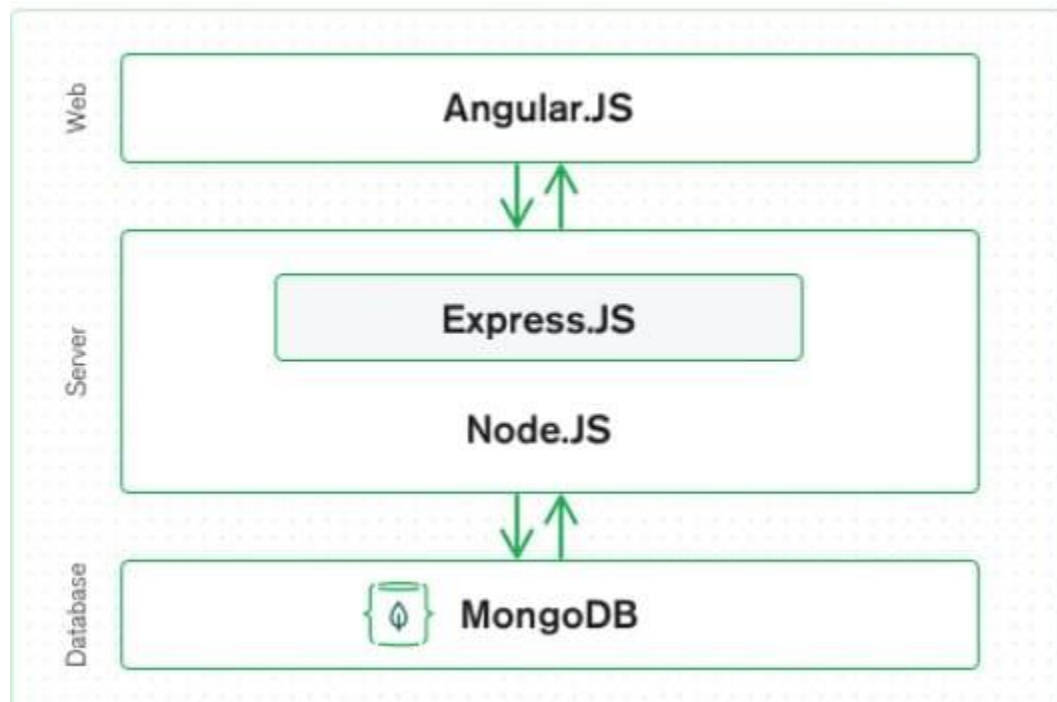
Middleware es software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él. Básicamente, funciona como una capa de traducción oculta para permitir la comunicación y la administración de datos en aplicaciones distribuidas. A veces, se le denomina “plumbing” (tuberías), porque conecta dos aplicaciones para que se puedan pasar fácilmente datos y bases de datos por una “canalización”. El uso de middleware permite a los usuarios hacer solicitudes como el envío de formularios en un explorador web o permitir que un servidor web devuelva páginas web dinámicas en función del perfil de un usuario.

¿Cuál es el significado del término “MEAN stack”?

Stack Mean está basado en Javascript para desarrollar aplicaciones web. Mean lleva el nombre de M-mongoDB, E-express, A-angular y N-nodejs, que corresponden a las cuatro tecnologías claves que componen las capas del Stack.

- MONGODB: base de datos.
- Express: framework de node.js.
- Angular: framework de Javascript del lado del cliente.
- NodeJs: principal servidor web de Javascript.

La arquitectura Mean está diseñada para hacer que la creación de aplicaciones web en Javascript y el manejo de JSON sean bastante fáciles.



¿Cuáles son las características/funcionalidades que Express ofrece como middleware?

Las funciones de middleware pueden realizar las siguientes tareas:

- Ejecutar cualquier código.
- Realizar cambios en la solicitud y los objetos de respuesta.
- Finalizar el ciclo de solicitud/respuestas.
- Invocar la siguiente función de middleware en la pila.

Si la función de middleware actual no finaliza el ciclo de solicitud/respuestas, debe invocar `next()` para pasar el control a la siguiente función de middleware. De lo contrario, la solicitud quedará colgada.

¿Qué es una ruta en express?

El direccionamiento hace referencia a la determinación de cómo responde una aplicación a una solicitud de cliente en un determinado punto final, que es un URI (o una vía de acceso) y un método de solicitud HTTP específico (GET, POST, etc.).

Cada ruta puede tener una o varias funciones de manejador, que se excluyen cuando se correlaciona la ruta.

La definición de ruta tiene la siguiente estructura:

`app.METHOD(PATH, HANDLER)`

Donde:

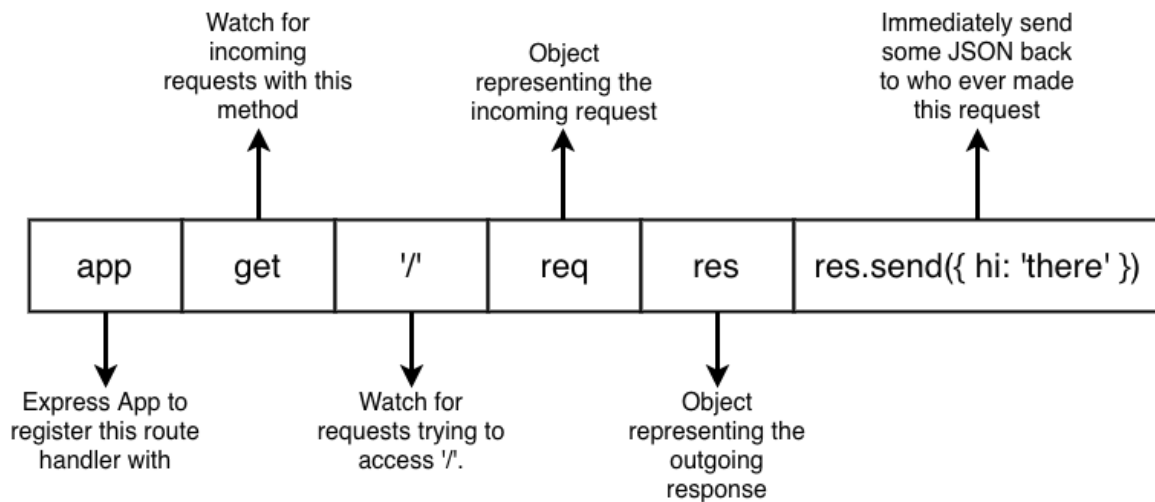
- `app` es una instancia de express.
- `METHOD` es un método de solicitud HTTP.
- `PATH` es una vía de acceso en el servidor.
- `HANDLER` es la función que se ejecuta cuando se correlaciona la ruta.

¿Qué es un “route handler” en express y para qué se utiliza?

Un route handler en Express se escribe de la siguiente manera:

```
app.get('/', (req, res) => {  
  res.send({  
    hi: 'there',  
  });  
});
```

Y así es como lo interpreta Express internamente:



<u>The Request Wants To...</u>		
app	get	Get info
	post	Send info
	put	Update all the properties of something
	delete	Delete something
	patch	Update one or two properties of something

Inmediatamente después de los route handlers tendremos una línea como esta, donde Express le indica a Node que escuche tráfico entrante por el puerto 5000:

```
app.listen(5000);
```

¿Cuáles son los 2 parámetros necesarios para crear una ruta para el método GET con express, ejemplo: `app.get()`?

Se necesitan los parámetros `req` y `res`, además se utiliza la función `listen` como se muestra en el ejemplo:

```
app.listen(3000); //Puerto por donde escucha
app.get('/uno', (req, res) => { //route handler
  res.send('Hello World, from route one');
});
```

¿Qué es lo que hace la instrucción `res.send`?

Envía una respuesta, puede ser de distintos tipos como HTML por ejemplo:

```
app.get('/prueba', (req, res) => { //route handler
  res.send('<h1> Hello world, from route Prueba </h1>' );
});
```

¿Cuál es el motivo por el que express es tan popular?

La razón por la que Express es el marco web más popular es que facilita el desarrollo de aplicaciones web, sitios web y API. También ofrece una colección subyacente de topografías. Con Express.js, podrá perfeccionar diferentes aspectos de la aplicación web. Puede determinar configuraciones como la ubicación de las plantillas que se usarán para la respuesta o el puerto que se usará para establecer una conexión.

Ofrece además estas características:

- Un marco de trabajo de código abierto
- Se centra en el alto rendimiento
- Cobertura de prueba súper alta
- Admite múltiples motores de plantillas (lo que simplifica la generación de HTML)
- Le permite escribir respuestas a URL específicas
- Un mecanismo simple que localiza errores en las aplicaciones rápidamente

Referencias

npm: express. (s. f.). Npm. Recuperado 20 de marzo de 2022, de <https://www.npmjs.com/package/express>

Qué es middleware: definición y ejemplos. (s. f.). Microsoft Azure. Recuperado 21 de marzo de 2022, de <https://azure.microsoft.com/es-mx/overview/what-is-middleware/>

Direccionamiento de Express. (s. f.). express. Recuperado 21 de marzo de 2022, de <https://expressjs.com/es/guide/routing.html>

Direccionamiento básico de Express. (s. f.). express. Recuperado 21 de marzo de 2022, de <https://expressjs.com/es/starter/basic-routing.html>