

Desarrollo Web Full Stack Node

Ejercitación - M03C14a

Middleware

Introducción

En esta ocasión te pedimos dejar de lado los proyectos que hasta el momento has venido trabajando y te concentrés en el trabajo con los middlewares. Pues los mismos son parte fundamental de una aplicación de Express y es necesario que comprendás cómo trabajar con ellos y así tener el control general de tu aplicativo.

Para tal finalidad, te dejamos [estos archivos base](#) para que puedas llevar a cabo la ejercitación.

Objetivo 1 - Log Middleware

Se busca que dentro del aplicativo exista un archivo de texto llamado **userLogs.txt**, el mismo deberá estar ubicado en una carpeta con nombre **logs**. Dicho archivo, deberá registrar el ingreso a cualquier ruta del sistema. Cada registro deberá quedar así:

El usuario ingresó a la ruta: _____

Para ello será necesario que crees una carpeta **middlewares** y dentro de ella el middleware llamado **userLogs.js**.

Tené en cuenta que este middleware se usará en toda la aplicación, por lo tanto deberás implementarlo según como corresponda. Y así mismo podrás crear las rutas que estimes necesarias para probar el middleware.

Algo importante que vale la pena destacar, es que el archivo **userLogs.txt** deberá ir almacenando **TODAS** las rutas a las que accede el usuario. El mismo deberá verse así:

El usuario ingresó a la ruta: /services

El usuario ingresó a la ruta: /services/design

El usuario ingresó a la ruta: /

...

Objetivo 2 - Admin Middleware

Ahora. Se busca que dentro del aplicativo exista una ruta así:

localhost:3000/admin?user=ABC

Dicha ruta deberá dejar ingresar al usuario, siempre y cuando el texto enviado en el query string como *user* sea exactamente: *Ada, Greta, Vim ó Tim*. Si se cumple lo anterior, se deberá enviar como response:

Hola Admin: Ada

De lo contrario el texto del response será:

No tienes los privilegios para ingresar

Lo anteriormente mencionado, **sólo deberá suceder para la ruta /admin.**

Para ello tené en cuenta al momento de crear la ruta, que la misma deberá responder a dos Callbacks. el 1ero será el middleware y el 2do el método que maneja el caso de ingreso exitoso. Ejemplo:

router.get("/admin", middleware, método);

Sin más que agregar, llegó tu momento de *codear y romper tus barreras mientras lo hacés.*

¡Mucha suerte!