



NODE.JS



Luis Miguel Avendaño Lozano
ACADEMLO

TAREA 2

Temas:

- Dtos
- Variables de entorno
- Autenticación y autorización

Basado en el ejercicio de la semana pasada, nuestro senior developer nos hizo el siguiente comentario:

“Muy buen trabajo con tu solución de la semana pasada, estuve revisando tu código y creo que podemos optimizarlo

Debemos actualizar los modelos, te adjunto una imagen para que veas los cambios que hay que realizar.

Nuestros empleados también me comentaron que no hay una validación muy eficiente a la hora de levantar servicio o al crear usuarios, debemos implementar una mejor validación, creo que había escuchado de algo llamado DTOS

Antes de que se me olvide, asegúrate de poner nuestras credenciales de nuestra base de datos en un archivo de variables de entorno, pero estoy seguro que ya hiciste eso.

Ahora debemos asegurarnos que nuestros usuarios se sientan seguros al tener una cuenta con nosotros.

Por favor, asegúrate de encriptar las contraseñas de los usuarios, como ya sabrás, estas no pueden ser contraseñas planas, debemos hacerlas ilegibles para cualquier usuario que quiera robarnos nuestra información.

Junto con las contraseñas, porfavor implementa un endpoints para que el usuario tenga que iniciar sesión, y regresar un JWT para que pueda acceder a nuestros demás endpoints, solo los usuarios autenticados pueden hacer uso de nuestra app.

Junto con eso, implementa un middleware para validar el rol del usuario, ya que los empleados de nuestra compañía deberían ser los únicos para realizar ciertas acciones en la app, mas adelante te especifico los endpoints que necesito que protejas.

Por último, el equipo de front-end necesitan que los ayudes a implementarestas nuevas cosas, es decir, les podrías ayudar a que se envié el JWT de tu back.”

- Implementa los siguientes cambios en los modelos, y establece la relación entre ambos

| Users |
|----------|
| Id |
| name |
| email |
| password |
| role |
| status |

| repairs |
|--------------|
| Id |
| date |
| motorsNumber |
| description |
| status |
| userId |

vz

Instalar *dotenv* y aplicarlo para usar variables de entorno para nuestras credenciales de nuestra base de datos.

Instala **bcrypt** para encriptar las contraseñas

Implementa el siguiente endpoint

| <i>/api/v1/users</i> | | |
|----------------------|------------|--|
| <i>HTTP VERB</i> | <i>URL</i> | <i>Descripción</i> |
| POST | /login | Recibir por req.body (email, password), y validar que los datos sean validos |

Este endpoint debe regresar un JWT con el id del usuario como *payload*, posteriormente utilizarlo para validar la sesión del usuario.

Los siguientes endpoints deben ser protegidos para que únicamente el usuario con rol “**employee**” pueda hacer peticiones a estos:

| <i>/api/v1/repairs</i> | | |
|------------------------|------------|--|
| <i>HTTP VERB</i> | <i>URL</i> | <i>Descripción</i> |
| GET | / | Obtener la lista de motos pendientes (pending) de reparar |
| GET | /:id | Obtener una moto pendiente de reparar por su id |
| PATCH | /:id | Actualizar el status de una reparación ha completado (cambiar status a completed) |
| DELETE | /:id | Cancelar la reparación de un usuario (cambiar status a cancelled) |