

NESTLOVERS REPORT

ENDPOINTS

1. Endpoint auth

- @POST auth/register

Descripción: Este endpoint permite a un usuario registrarse en la aplicación.

Parámetros: Recibe un objeto 'RegisterDto' en el cuerpo de la solicitud, que contiene la información del usuario para crear una nueva cuenta.

Respuesta: Devuelve la información del usuario registrado.

- @POST auth/login

Descripción: El método de inicio de sesión es un controlador de ruta POST que recibe un objeto 'loginDto'. Este objeto contiene las credenciales del usuario para autenticar al usuario.

Parámetros: Recibe un objeto 'loginDto' con las credenciales del usuario a autenticar.

Respuesta: Devuelve la información del usuario y un token.

2. Endpoint users

- @POST users/assignRole/:userId

Descripción: Asigna un rol a un usuario. Valida que el rol sea un valor válido. Si el usuario no existe, arroja una NotFoundException.

Parámetros: Recibe el 'userId', que es el id del usuario a asignar el rol.

Recibe 'assignroleDto', que es el rol a asignar.

Respuesta: Devuelve el usuario con el nuevo rol.

- @PATCH users/update/:id

Descripción: Actualiza al usuario en la base de datos. Solo los usuarios con el rol 'SUPER_ADMIN' pueden actualizar cualquier usuario.

Parámetros: Recibe el 'id' del usuario a actualizar.

Recibe 'updateUserDto', que son los datos a actualizar.

Respuesta: Devuelve el usuario actualizado

- @DELETE users/remove/:id

Descripción: Elimina un usuario de la base de datos. Solo un usuario con el rol de 'SUPER_ADMIN' puede borrar usuarios.

Parámetros: Recibe el 'id' del usuario a borrar.

Respuesta: Devuelve el usuario eliminado.

3. Endpoint projects

- @POST projects

Descripción: Crea un nuevo proyecto. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta.

Parámetros: Recibe 'createProjectDto', que son los datos del proyecto a crear. Recibe el 'user' que crea el proyecto.

Respuesta: Devuelve el proyecto creado.

- **@GET projects**
Descripción: Devuelve todos los proyectos en la base de datos.
Respuesta: Devuelve todos los proyectos en la base de datos.
- **@GET projects/byID/:id**
Descripción: Devuelve todos los proyectos creados por un usuario. La ruta está protegida y solo los usuarios con rol 'ADMIN' pueden acceder a esta.
Parámetros: Recibe el 'id' del usuario.
Respuesta: Devuelve todos los proyectos creados por un usuario de acuerdo con su id.
- **@GET projects/own**
Descripción: Devuelve todos los proyectos creados por el usuario logueado. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta.
Parámetros: Recibe el 'user' que está logueado.
Respuesta: Devuelve todos los proyectos creados por el usuario logueado.
- **@PATCH projects/update/:id**
Descripción: Actualiza un proyecto por su id. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta. El usuario debe ser el dueño del proyecto para actualizarlo.
Parámetros: Recibe el 'id' del proyecto. Recibe 'updateProjectDto', que son los datos del proyecto a actualizar.
Respuesta: Devuelve el proyecto actualizado.
- **@DELETE projects/delete/:id**
Descripción: Elimina un proyecto por su id. La ruta está protegida y solo los usuarios con rol 'SUPER_ADMIN', esto es porque solo los SUPER_ADMIN pueden borrar cualquier proyecto.
Parámetros: Recibe el 'id' del proyecto.
Respuesta: Devuelve el proyecto eliminado.
- **@DELETE projects/deleteMy/:id**
Descripción: Elimina un proyecto por su id. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta. El usuario debe ser el dueño del proyecto para poder borrarlo.
Parámetros: Recibe el 'id' del proyecto.
Respuesta: Devuelve el proyecto eliminado.

4. Endpoint documents

- **@POST documents/docPhase1Part1/project/:id**
Descripción: Crea un nuevo documento correspondiente a la Fase 1, Parte 1 para un proyecto específico.
Parámetros: Recibe 'createDocumentDto', que es el DTO que contiene los datos necesarios para crear el documento. Incluye el contenido del

documento. Recibe 'projectId', que es un identificador único del proyecto al que pertenece el documento.

Respuesta: Devuelve el documento creado.

- @GET documents
Descripción: Devuelve todos los documentos en la base de datos.
Respuesta: Devuelve todos los documentos en la base de datos.
- @GET documents/byID/:id
Descripción: Devuelve un documento por su id.
Parámetros: Recibe el 'id' del documento.
Respuesta: Devuelve el documento dado su id.
- @GET documents/byProject/:id
Descripción: Devuelve todos los documentos que pertenecen al proyecto. La ruta está protegida y solo los usuarios con rol 'ADMIN' pueden acceder a esta.
Parámetros: Recibe el 'projectId', que es el id del proyecto.
Respuesta: Devuelve todos los documentos que pertenecen al proyecto.
- @GET documents/own/:id
Descripción: Trae de vuelta los documentos que pertenecen al usuario actual para un proyecto específico. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta. El usuario debe ser el dueño del proyecto para obtener los documentos.
Parámetros: Recibe el 'projectId', que es el identificador único del proyecto para el cual se van a traer de vuelta los documentos.
Respuesta: Devuelve un arreglo de los documentos que pertenecen al usuario actual del proyecto especificado.
- @PATCH documents/update/:docId
Descripción: Actualiza un documento por su id. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta. El usuario debe ser el dueño del documento para poder actualizarlo.
Parámetros: Recibe el 'id' del documento. Recibe 'updateDocumentDto', que son los datos a actualizar.
Respuesta: Devuelve el documento actualizado.
- @DELETE documents/delete/:id
Descripción: Elimina un documento por su id. La ruta está protegida y solo los usuarios con rol 'SUPER_ADMIN', esto es porque solo los SUPER_ADMIN pueden borrar cualquier documento.
Parámetros: Recibe el 'id' del documento.
Respuesta: Devuelve el documento eliminado.
- @DELETE documents/deleteMy/:docId

Descripción: Elimina un documento por su id. La ruta está protegida y solo los usuarios con rol 'ADMIN' o 'USER' pueden acceder a esta. El usuario debe ser el dueño del documento para poder eliminarlo.

Parámetros: Recibe el 'docId', que es el id del documento.

Respuesta: Devuelve el documento eliminado.

Autenticación, autorización y persistencia en la base de datos.

Para la autenticación decidimos usar JWT (JSON web token), esto para que cuando registremos nuestra información de usuario la podamos verificar después del logueo. Es decir, cuando registramos nuestra información de usuario esta queda guardada en la base de datos, luego se compara con las credenciales que deberían estar guardadas en la DB, si la contraseña coincide con la digitada, entonces se creará un JWT que contendrá el email, id y rol del usuario. Este JWT se guardará en una white list en la base de datos, cuando se haga logout, este mismo JWT se eliminará.

Para la autorización usamos algo muy parecido al autenticación, haciendo que el JWT autorice los permisos necesarios de cada usuario dependiendo de las acciones correspondientes a su rol en el programa.

Finalmente para la parte de persistencia en la base de datos usamos Type ORM junto con una BD de postgre en la nube y desplegada con Neon DB. El uso de las tablas fueron definidos por entidades en los diferentes módulos.