

Informe Backend NestJs

Estructura General

- **Módulo de Autenticación:** Gestiona la autenticación y el manejo de usuarios.
- **Módulo de Ebooks:** Gestiona las operaciones relacionadas con ebooks.
- **Módulo de Payment:** Gestiona los pagos por los ebooks con payu.
- **Módulo Principal:** Módulo central que integra otros módulos.

Autenticación

Se utilizó JWT para la autenticación. El AuthService se encarga de la validación de usuarios, generación de tokens y registro de usuarios. Interactúa con la entidad User para realizar consultas relacionadas con usuarios.

Se implementó una estrategia usando JWT que está definida en la clase JwtStrategy, que es parte del módulo de autenticación. Autentica a los usuarios basándose en el JWT pasado en los encabezados de la solicitud.

Autorización

Los usuarios se diferencian por roles (Admin, Author, Reader), que se utilizan para autorizar diferentes acciones de la API basadas en el rol asociado con el JWT. Se creó un Guard que verifica los roles que tienen acceso a determinados métodos HTTP, que están indicados usando un decorator Roles.

Persistencia de Base de Datos

Se utilizó TypeORM como una herramienta ORM para interactuar con la base de datos PostgreSQL. El ORM ayuda a mapear a tablas de base de datos las entidades como User, Reader, Author, Admin, Ebook. Cada tipo de usuario (Reader, Author, Admin) extiende una entidad base User que maneja propiedades comunes como username, password y email.

Funcionalidades Implementadas:

1. **Registro de usuarios:** Los usuarios pueden registrarse en la aplicación como usuarios lectores o usuarios autores.
2. **Inicio de sesión:** Después de registrados los usuarios pueden acceder por medio de un login, que permite la autenticación.
3. **Edición de usuario:** Los usuarios pueden modificar la información de su cuenta.
4. **Eliminación de usuario:** Los usuarios pueden eliminar sus cuentas si así lo desean.
5. **Registro de ebooks:** Los usuarios autores después de autenticarse pueden publicar ebooks que quedan registrados a su nombre.
6. **Eliminación de ebook:** Los autores pueden eliminar cualquier libro que hayan registrado si así lo desean.
7. **Compra de ebook:** Los usuarios pueden comprar cualquier libro que se encuentre en la plataforma.
8. **Lista de ebooks disponibles:** Los usuarios pueden consultar todos los ebooks disponibles en la plataforma.

9. **Lista de ebooks comprados:** Los usuarios pueden consultar todos los ebooks que han comprado en la plataforma.
10. **Selección de libros deseados:** Los usuarios pueden seleccionar ebooks que sean de su gusto y guardarlos en una lista de deseados.
11. **Lista de ebooks deseados:** Los usuarios pueden consultar los libros que se encuentran en su lista de deseados.
12. **Sistema de pagos:** Los usuarios autores pueden añadir una cuenta la que se consignarán las ganancias por libros
13. **Sistema de compra:** Los usuarios pueden comprar un ebook de manera digital haciendo uso de la pasarela de pagos de payu.
14. **Filtro de ebook:** Los usuarios pueden filtrar las búsquedas de los ebook por categorías, autor, precio o cantidad de likes.

Endpoints de la API:

Módulo de Autenticación

Método HTTP	Ruta	Parámetros	Respuesta	Descripción
POST	/auth/register	Dto de creación de usuario.	JWT	Registra un nuevo usuario
POST	/auth/login	Dto de inicio de sesión de usuario.	JWT	Autentica a un usuario
GET	/auth/status	Request	JWT	Verifica y devuelve el estado actual de autenticación de un usuario
GET	/auth/:id	Id de un usuario	Usuario	Recupera detalles del usuario por ID
GET	/auth	N.A.	Lista de usuarios	Recupera detalles de todos los usuarios
PATCH	/auth/:id	Id de un usuario y Dto de edición de usuario	Usuario	Actualiza detalles de un usuario existente
DELETE	/auth/:id	Id de un usuario	DeleteResult	Elimina un usuario existente
GET	/auth/session	Request	Usuario logueado	Obtiene el usuario que se encuentra logueado

Módulo de Ebooks

Método HTTP	Ruta	Parámetros	Respuesta	Descripción
POST	/ebooks	Dto de creación de ebook.	Ebook	Registra un nuevo ebook
GET	/ebooks/info/:id	Id de un ebook	Dto de ebook con solo información básica.	Recupera detalles del ebook por ID
GET	/ebooks/visualize/:id	Id de un ebook	Dto de ebook con datos del archivo	Recupera archivo del ebook por ID
GET	/ebooks	N.A.	Lista de ebooks	Recupera detalles de todos los ebooks
PATCH	/ebooks/:id	Id de un ebook	Ebook	Actualiza detalles de un ebook
DELETE	/ebooks/:id	Id de un ebook	DeleteResult	Elimina un ebook
POST	/ebooks/ebookReader	Dto de creación de objeto que relaciona los ebook con el usuario que lo ha comprado	Ebook Reader	Añade una relación entre un usuario y un libro.
GET	/ebooks/:readerId	Id de usuario	Lista de ebooks	Obtiene todos los ebooks que tiene un usuario

Módulo de Order

Método HTTP	Ruta	Parámetros	Respuesta	Descripción
POST	/order/create	Dto de creación de orden	String	Crea una order
GET	/order/:orderId	Id de una orden	Order	Recupera una order por ID
GET	/order/	Request	Lista de order	Recupera todas las order de un usuario logueado.
GET	/order/:userId/orderId	Id de un usuario y Id de una orden	Order	Recupera una order específica de un usuario específico

GET	/order/all	N.A.	Lista de order	Recupera todas las order existentes
GET	/order/:id	Id de una order	Order	Recupera una order dado un ID
POST	/order/create/:id	Id	String	Le crea una orden a un usuario en específico

Módulo de Payment

Método HTTP	Ruta	Parámetros	Respuesta	Descripción
GET	/payment/transactions/:transactionsId	Id de transacción	Transaction	Recupera una transacción por Id
GET	/payment/transactions	Request	Lista de transaction	Recupera todas las transacciones
GET	/payment/payu-response	Dto de transacción y Response	Json de la transacción creada	Crea una transacción después de una transacción terminada cuando el usuario le hunde volver a la página
GET	/payment/payu-confirmation	Dto de transacción y Response	Json de la transacción creada	Crea una transacción después de una transacción terminada cuando la BD de payu anuncia el estado asincrónicamente
GET	/payment	N.A	Lista de transacciones	Recupera todas las transacciones existentes
GET	/payment/:id	Id de transacción	Transaction	Recupera una transacción por Id

Detalles de Implementación

- Cada módulo tiene su propio servicio y controlador que manejan la lógica de negocio y los endpoints de la API, respectivamente.
- Los servicios utilizan inyección de dependencias para acceder a las instancias del repositorio de entidades.

- Las contraseñas se cifran utilizando bcrypt antes de almacenarse en la base de datos. De la misma manera se usa para hashear y verificar contraseñas, aumentando la seguridad durante la autenticación.
- El sistema de pagos integrado en nuestra aplicación se gestiona a través de PayU, una reconocida pasarela de pagos que facilita las transacciones electrónicas. En nuestra API, el módulo de Payment se encarga de interactuar con PayU para procesar compras y manejar la confirmación de las transacciones.

Descripción del Entorno de Producción

URL de la Aplicación Desplegada: Nuestra aplicación backend está desplegada en el entorno de Render en la siguiente URL: <https://backend-grupo-13-deploy.onrender.com/>. Esto permite a cualquier usuario acceder a la funcionalidad de la aplicación en un entorno real.

Base de Datos: La base de datos asociada a nuestra aplicación está alojada en ElephantSQL.

Proporciona un servicio de base de datos gestionada que garantiza la confiabilidad y el rendimiento.

Cobertura de Pruebas

Cobertura de Pruebas del 70%: Nuestro conjunto de pruebas actual proporciona una cobertura del código del 70%. Esto significa que la mayoría de las funcionalidades de la aplicación están respaldadas por pruebas automatizadas que verifican su correcto funcionamiento. Las pruebas son esenciales para garantizar la estabilidad y la confiabilidad de la aplicación a lo largo del tiempo.