

PRÁCTICA EVALUABLE– 2º DAW DWES 10/12/2024

La empresa de libros Álvaro Marín necesita una aplicación para gestionar stock de los libros que tiene a disposición en sus numerosas tiendas físicas. En la parte de backend, se solicita la creación de una API Rest con la siguiente estructura de entidades y una serie de funcionalidades.

Se disponen de 3 entidades principales que son: BOOK, STORE y STOCK. La entidad BOOK almacena la información de un libro, Store almacena la información de una tienda de la compañía que puede ser una tienda física o un almacén.

Book:

- id (BIGINT) – Autoincremental
- isbn(String) Ej: 0-111-222-3
- type (Enum) - DRAMA, SCIFI, MANGA, SPORT, COOKING...
- status (Enum) – ONSALE, PREORDER
- price (Double) - Valor del libro
- published (LocalDate) - Fecha de publicación Ej: 2024-02-02
- Fecha de modificación y creación

Store:

- id (BIGINT) – Autoincremental
- dir(String) - Dirección
- type (ENUM) – STORE, PHYSICAL, OTHER
- sale (Boolean) – Venta en el lugar
- Fecha de modificación y creación

Stock:

- id (BIGINT) – Autoincremental
- amount(Integer) - cantidad
- Book @JoinColumn
- Store @JoinColumn
- Fecha de modificación y creación

Para modelar la aplicación, se usará la arquitectura por capas (Controller-Service-Repository). El almacenamiento se realizará en una BBDD MYSQL. Definir un conjunto de datos de book, store y stock. Para la carga de datos, usar seeders para poblar con valores las tablas.

Funcionalidades - Controller	Entrada	Salida	HTTP code
Listado de libros en ONSALE (Se muestran los libros cuyo estado es ONSALE)		List<Book>	200-OK
Crear libro	Book	Book	201- CREATED
Consultar stock de un libro (Se indica donde hay stock y la cantidad de un libro)	idBook	List<ResponseStock> {amount, Store }	200 - OK 404 - NOT FOUND (No existe libro)
Actualizar stock de un libro de una tienda (Se actualiza el amount de un libro de una tienda indicada)	idBook, amount, idStore		200 - OK 404 - NOT FOUND (No existe libro o tienda)
Buscar libro por ID (Se devuelve la información del libro indicado)	bookId	Book	200 - OK 404 - NOT FOUND (No existe libro)
Eliminar un libro (Se elimina el libro indicado y su información asociada de BBDD)	bookid		200 - OK 404 - NOT FOUND (No existe libro)

Crea una excepción que será capturada en el fichero Bootstrap/app cuando no se encuentra un libro. En la sección Exceptions.

Crea una colección en thunderclient o postman de las peticiones, exportar dicha colección y entregarla en la tarea.

Para la nota se tiene en cuenta la arquitectura, funcionalidad implementada y codificación.