



Programacion Concurrente - Biblioteca

David Tinaut Martínez

November 27, 2023

Introducción

En esta práctica se pretende utilizar el framework **Spring** para poder elaborar una aplicación web en Java que simule una gestión de una biblioteca, el usuario podrá interactuar y podrá agregar lectores, bibliotecarios, préstamos y libros, todas estas entidades con la ayuda de Hibernate, que es un framework especializado para las primeras dos capas (la capa de datos DTO y la capa de acceso a los datos DAO). Hibernate (DB Generation) funciona mediante el marcado de clases mediante anotaciones para realizar un mapeo entre nuestras entidades POJO en Java y nuestras tablas en SQL. En este caso he utilizado MySQL para almacenar estas operaciones básicas CRUD en mi aplicación.

Desarrollo

Para poder realizar la estructura de Spring, he utilizado la página web <https://bootify.io/> para que pueda hacerme todo el empaquetado de dependencias Maven que voy a utilizar, la versión de Spring, habilitar Docker, lenguaje Java o Kotlin, un frontend con Thymeleaf + Bootstrap + Webpack, y una version Java 17 algo más antigua pero más estable para evitar problemas de fase de pruebas o que aún no se han implementado. Añadiré las dependencias: -SPRING WEB. -HATEOAS. -HIBERNATE. -JPA. -MUSTACHE. -LOMBOK. -PERSISTENCE. -MYSQL-CONNECTOR. Una vez añadidas las dependencias, toca crear la estructura de las entidades y las relaciones de la base de datos, una superclase Biblioteca controlará el resto como capa Domain, y el resto Layer tendrá una serie de relaciones entre ellas, como por ejemplo que un Bibliotecario puede ser un Lector,

pero el Lector jamás será un Bibliotecario, una vez definida la estructura se generará el proyecto y será importando al IDE que utilizo por defecto, que es IntelliJ para comenzar a trabajar:

Base de Datos

Una vez dentro, arrancaré el main de Spring Application, arrancará un servidor Tomcat en el puerto 8080 para poder ver la estructura de la Web y arrancar todo el servicio. Para poder terminar de ajustar la configuración del proyecto, agregaré en application.yml la base de datos, mi usuario y contraseña. junto con el puerto de MySQL 3306 por defecto conseguiré una conexión con la base de datos, por último falta utilizar y conectar un contenedor Docker.

Docker

Docker se encarga de automatizar el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de la virtualización de aplicaciones en múltiples sistemas operativos. Una vez que he desplegado Docker en mi máquina, mapeará los puertos 3307 y 3306, creará un contenedor y un volumen para almacenar la información de la base de datos mediante el archivo docker-compose.yml con las credenciales de mi base de datos.

Problemas Encontrados

Una vez que almaceno cualquier tipo de Entidad en la base de datos, Docker utiliza el volumen de archivo creado para almacenar la información, aunque creo que también debería de almacenarse en mi conexión MySQL que he abierto con IntelliJ usando las credenciales en forma de tablas. Otro problema que no supe solucionar fue la parte vista de la aplicación con los scss, intenté maquillarlo un poco con el html.

Referencias

1. <https://maven.apache.org/guides/index.html>
2. <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
3. <https://www.thymeleaf.org/documentation.html>
4. <https://webpack.js.org/concepts/>
5. <https://docs.npmjs.com/>
6. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

7. <https://htmx.org/docs/>