Taller 2

Intento de Documentación

Bueno jaja tin.

Empecemos por la gestión de los datos dentro de H2 (La base de datos de Spring-Boot). Primero toca configurarla entonces melo. En la carpeta **resources** va a estar el archivo de **application.properties** y ahí es donde se configura la base de datos con los siguientes comandos:

spring.datasource.url=jdbc:h2:file:C:\Users\Sergio Duque\OneDrive\Desktop\SAMUEL\Programar\Proyecto\_Taller2/mi\_db;AUTO\_SERVER=TRUE

//Este sirve para crear la URL de la bd (con el url=jdbc) después se establece la ruta donde se va a guardar la bd con h2:file:ruta\_deseada (si se quisiera crear una bd no persistente, es decir, que se guarde en la memoria, se define así h2:mem). Y finalmente el mi\_db; que hace que se cree una carpeta con ese nombre en la ruta mencionada. Y el AUTO\_SERVER=TRUE Permite que la base pueda ser usada por varios procesos (útil si accedes desde la consola H2 y desde tu app al mismo tiempo).

spring.datasource.driverClassName=org.h2.Driver

//Indica la clase del driver, normalmente este comando sobra ya que implícitamente ya está indicada, pero se pone por si las.

//Usuario y contraseña para la datasource, se pone “sa” de usuario ya que esta no tiene contraseña por default entonces tin, de momento eso va.

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

//Esto es pa cuando cambies entidades, no se borren datos sino que se creen y/o actualicen columnas y/o filas.

spring.jpa.show-sql=true

//Esto sirve pa que en la consola muestre que ejecuta hibernate, o sea, select/update/delete…etc.

spring.h2.console.enabled=true

//Activa la consola web para ver las tablas y en sí poder ver toda la bd en el navigaytor

spring.h2.console.path=/h2-console

//Y esto es pa darle la ruta en el navegador, por default sería http://localhost:8080/h2-console

Melo, ya con esos comandos ya configuró la BD. Que calidad.

Ya rey empiece con los modelos y los DAO.

Usma nos enseñó los DAO de una manera toda arcaica 🙃. Pero bueno tin, lo que toca hacer de la forma que dice Usma, es crear una interfaz y después crear una clase que la implemente entonces what a shit. Pero pues eso se puede hacer con una sola interfaz gracias al Spring-Boot.

En vez de crear la interfaz y la clase, se usa solamente el “extends JpaRepository<>” y ya con eso tenés la biblioteca con todos los métodos sin necesidad de un Entity Manager, es más rápido, pero hay que tener en cuenta que de la forma arcaica se aprende más y además hay más control sobre las consultas exóticas.

-Métodos de JpaRepository

save(entity) // Inserta o actualiza un registro

findById(id) // Busca por ID

findAll() // Devuelve todos los registros

deleteById(id) // Elimina por ID

delete(entity) // Elimina una entidad

count() // Devuelve el número de registros

existsById(id) // Verifica si existe por ID