CRUD BASICO Angular, Spring boot y MySql

* Funciona todo.
* He seguido los tutoriales como la base, pero mi proyecto está muy cambiado respecto del que propone el autor.
* Agregué una cantidad de comentarios.
* Cambié a una estructura de paquetes más estándar.
* La implementación que hace el autor del método subscribe() está deprecada. La cambié por la implementación que ahora se recomienda.
* JPA puede crear la base de datos y las tablas, y cargar datos. Por ahora nos quedamos con el concepto, pero no lo vamos a ver en detalle.

# #1 CRUD BASICO Angular, Spring boot y MySql: creación de proyecto y modelo en backend

<https://youtu.be/4VWQwrM1voU>

CRUD básico (create-read-update-delete) full-stack con Spring Boot como backend, Angular 12 como frontend y MySQL como base de datos.

El proyecto es un administrador de tareas, o tasks.

Arranco WampServer, para la base de datos.

Base de datos: tareas

El archivo Database tareas.sql tiene el script para crear la base de datos. Notar que no es igual al que usa el autor. Le hice unos cambios.

En MySQL no existe un tipo de dato bool o boolean. Los acepta como provenientes de otros DBMS y los representa como tiny int(1).

<https://stackoverflow.com/a/42147995/2740402>

For the record, the spring.jpa.hibernate.ddl-auto property is Spring Data JPA specific and is their way to specify a value that will eventually be passed to Hibernate under the property it knows, hibernate.hbm2ddl.auto.

The values create, create-drop, validate, and update basically influence how the schema tool management will manipulate the database schema at startup.

For example, the update operation will query the JDBC driver's API to get the database metadata and then Hibernate compares the object model it creates based on reading your annotated classes or HBM XML mappings and will attempt to adjust the schema on-the-fly.

The update operation for example will attempt to add new columns, constraints, etc but will never remove a column or constraint that may have existed previously but no longer does as part of the object model from a prior run.

Typically in test case scenarios, you'll likely use create-drop so that you create your schema, your test case adds some mock data, you run your tests, and then during the test case cleanup, the schema objects are dropped, leaving an empty database.

In development, it's often common to see developers use update to automatically modify the schema to add new additions upon restart. But again understand, this does not remove a column or constraint that may exist from previous executions that is no longer necessary.

In production, it's often highly recommended you use none or simply don't specify this property. That is because it's common practice for DBAs to review migration scripts for database changes, particularly if your database is shared across multiple services and applications.

start.spring.io

Maven

Java

2.6.5

gga

tareas

Administrador de tareas

Jar

11

Spring Web

Spring Data JPA

Spring Boot DevTools

MySQL Driver

Genero y descargo.

Editar application.properties de modo que quede

spring.datasource.driver-class-name=com.mysql.cj.jdbc.driver

spring.datasource.url=jdbc:mysql://localhost:3306/tareas

spring.datasource.username=root

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=none

logging.level.org.hibernate.SQL=debug

# #2 CRUD BASICO Angular, Spring boot y MySql: service y controller

<https://youtu.be/icqYtxxSpEY>

# #3 CRUD BASICO Angular, Spring boot y MySql: Angular Creacion de proyecto y service

<https://youtu.be/3divgzHPX5g>

# #4 CRUD BASICO Angular, Spring boot y MySql: diseño de formularios con bootstrap y rutas

<https://youtu.be/Z6zcyenQSwY>