Spring Boot Application with Data JPA - It's a web Service which interacts to any SQL based Database. (RDBMS)

We created a Spring boot Application (Starter Project - Parent project)

1) Spring Web

2) Spring Data JPA

3) Springboot Dev Tools

4) H2, MySQL, MSSQL, Postgres (Different DB driver)

5) Lombok

Must, Should & could have packages.

1) Entity Bean Class

2) Repo Interface

3) Service (Business Logic)

4) Controller (End point mappings)

Annotations used in Spring Boot

@SpringBootApplication = @EnableAutoConfiguration + @SpringBootConfiguration + @ComponentScan

@Autowired - This is will help to inject the Bean at runtime.

@Service

@Repository

@Component

JPA Annotations

@Entity

@Table

@Column

Lombok based Annotations

@Data

@NoArgsConstructor

@AllArgsConstructor

@Getter

@Setter

@EqualsAndHashCode

@Builder

@NonNull

@Slf4J

@ReqArgsConstructor

Tested all the End-points using

1) SoapUI

2) Swagger UI (API Documentation)

3) Curl (CLI Utility = swagger uses curl to interact with rest end-points)

Some Sample Data to the DB

Stored sample data in the following files within resource folder

schema.sql

data.sql

spring.application.name=mydatajpa

server.port=8085

# H2 Properties (Comment in properties file)

#spring.datasource.url=jdbc:h2:mem:sampleh2db

#spring.datasource.username=sa

#spring.datasource.password=

#spring.datasource.driver-class-name=org.h2.Driver

spring.jpa.hibernate.ddl-auto=none

#update - for persistence DB like mysql,postgres, ms-sql

#none - for in-memory or embedded db

spring.jpa.show-sql=true

#MySQL DB Properties

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

spring.datasource.username=root

spring.datasource.password=root

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

Agenda

Best Practices in Spring Boot Application Development

1. Proper End point naming conventions
2. API Versioning – api/v1/users
3. Proper naming conventions for variables, method and classes
4. Adequate Test coverage [Unit, integration & e2e Testing)

CRUD Operation [POST/GET/PUT/DELETE]

@PathVariable

@RequestBody (This is created by client & end-user)

@ResponseBody with the status code (Created by server)

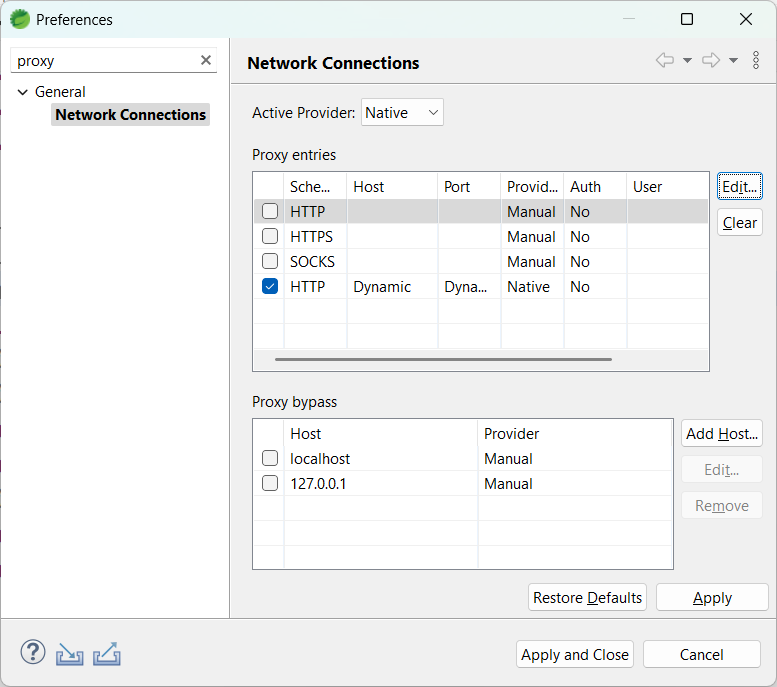
Add Lombok.jar to Eclipse & STS root folder.

Copy the jar and paste it inside the “plugins” folder of Eclipse &/ STS.

Restart the IDE after adding jars.

Proxy Server config in Eclipse or STS

Window 🡪 Preferences (Search for proxy)



JPA – Java Persistence API

Persistence – Storage of Data in DB (Ability to store the data in DB)

Storing the State of Objects in Flat file – Serialization (Stores the data in encrypted)

Storing the state of objects in DB Entity (Table) – Persistence.

RAM & ROM (Volatile)

Random Access Memory (Volatile)

Read Only Memory (Non-Volatile)

JDBC – API ( Specifications) -- Driver (Implementation)

JPA – API (Specifications) – Set of Interface & Abstract Classes. Directly do the CRUD operations using Entity and related Methods CRUD – Save, find, Save, Delete

ORM – Object Relational Mapping (JAVA – RDBMS Connecting)

JDBC vs JPA

All 5 steps are Manual – Most of the Heavy lifting work will be taken care by the framework.

JPA is a Specification

JPA Implementations are

1. Hibernate Framework (HQL – Hibernate Query Lang – Independent of any RDBMS- Won’t use DB tables name, it uses just Java Entity Class names) hbm – hibernate mapping file (xml file) – Annotations
2. iBatis
3. MyBatis
4. EclipseLink
5. Spring Data JPA

Either specify entity bean class or DB table.

In Java everything is object, where as in RDBMS everything is Entity (Table, Stored Procedure, Sequence, Trigger, Materialized View, functions, Constraints etc.,)

Pathvariable -- api/v1/trainers/10

RequestParam (RequestParameters google.com?q=”what is java”&lang=us-en)

Key-value pairs

<https://www.geeksforgeeks.org/advance-java/configuring-store-procedure-in-spring-boot-application/> -- Spring Data JPA invoking Stored Procedure

Week2 Revisit

Java 8 New Features

LTS – Long Term Support Version

Functional Interface – An Interface with only one abstract method (0 or more default & static method) [java.util.function] - Supplier, Predicate, Function, Consumer

1. Using Lambda (Anonymous [nameless] method => )
2. Using Method Reference ::

Streams (Flow of data – enable functional programming) – Multiple operations in a single line [ intermediate operation & terminal operations]

New Date & Time API (localDate, LocalDateTime)

NIO

Spring Boot –

IoC (Inversion of Control) & DI (Dependency Injection)

Web Service

SOA – Service Oriented Architecture

SOAP & REST ful.

Spring Framework – It’s a popular open source Java based framework used to create loosely coupled enterprise application.

Various modules in Spring framework – Spring Core, Spring Data, Spring Security, Spring AOP, Spring AI, Spring Web, etc.,

Tightly Coupled vs Loosely coupled (Spiderman toy vs Lego Spiderman toy)

Spring Boot – Simplified method of creating spring based projects.

Spring Data JPA –

Various Annotations.

URL vs URI

Monolith vs Microservice Arch.