Spring Data JPA – It’s JPA implementation of Spring Boot

Spring is very popular Java based framework

IoC – Inversion Of Control

DI – Dependency Injection

1. Loose coupling
2. Highly Scalable

Example --- Spiderman Toy -- Tight Coupling

Lego Kits – (Blocks) – Loosely coupled – that can be used to make any shapes

Let it be, car, bus, train, …..

Injection – Hospital ( Tablets, Syrups, )

Dependency Injection –

1. Constructor Injection -- Default/No Arg Constructor & All arg /fully parameterized constructor
2. Setter Injection – Def cons & Setter methods

To initialize an objects ---

Spring Initializr – Start.spring.io (To create a spring boot application using web interface/browser)

Full Stack

1. JAVA Full Stack (Spring Boot – Back End , Any JS based Framework – Front End, Postgres – Middle ware /DB
2. MERN Stack (MongoDB, ExpressJS, React, Node) – JSON
3. MEAN Stack (MongoDB, ExpressJS, Angular, Node) – JSON

POM

1. groupId (reverse of the company url/package name)
2. ArtifactId (project/application name)
3. Version (0.0.1-SNAPSHOT)

<https://spring.io/tools> - URL to download STS4

SpringBoot is called as Opinionated Framework

Application.properties (text based configuration)

In the form of key-value pairs

STS will give suggestion while adding the data in application.properties file

WebService – Service offered via internet (Web)

WebService – By using a uri executing a method which is defined in any programming lang

Web Service

1. SOAP – Simple Object Access Protocol (XML)
2. REST – Representational State Transfer (HTTP Methods - JSON)

End points – URI

URL – Uniform Resource Locator .html, .php,

URI – Uniform Resource Identifier (used in web services – won’t have any extension)

Bcos of Webservice, a .Net application can interact with Java application

A Java Application can interact with python application without any challengs

ABC Bank Customer (Debit Card)

1. ABC Bank’s ATM - (Deposit, withdraw, change pin, update address/email/Mobile, print last 5 transaction)
2. XYZ Bank’s ATM (deposit, withdraw, last 5 transactions)

JSON – Endpoint (debit card details, pin number)

H2 – InMemory Database (completely developed using JAVA)

@RequestingMapping – Generic Mapping End point annotation that can be used for any http method (get, post, put, delete, patch, trace, option)

@GetMapping – Specific Mapping

@PostMapping

Postman – Is a API Testing Tool

1. Download Standalone - <https://www.postman.com/downloads/>
2. Google Chrome - <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=en>

Annotations related Entity Bean class

1. @Entity
2. @Table (optional)
3. @Id
4. @Column (optional)
5. @OneToMany
6. @ManyToMany
7. @GeneratedValue

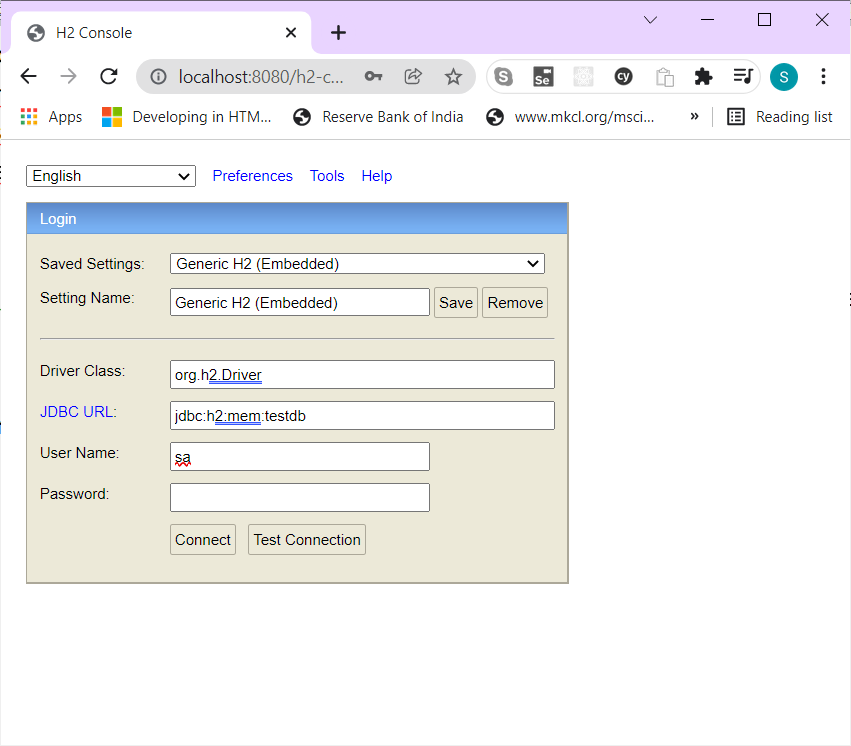
Ddl-auto = Automatically generate table if it is not available provided you have entity bean class

JPA –

Ddl-auto = create/update/validate/

jdbc:h2:mem:6f56caf4-5a02-4ff8-b061-dfc1d7f3d28a

localhost:8080/h2-console



jdbc:h2:mem:d1e458d9-0e83-4d04-b47c-c095870ef9a7

JDBC Driver = org.h2.Driver

JDBC url jdbc:h2:mem:testdb

username : sa

password :

REPO – DAO -Data Access Object (CRUD operation)

Create/Insert

Read

Update

Delete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CRUD Operation | Create/Insert | Read | Update | Delete |
| Http Method | Post | Get | Put | delete |
| Request Data | Object in Request Body | None or Id (Path Variable) | Id, Updated Object in Req Body | Id (Path Variable) |
| Annotations used | @PostMapping | @GetMapping | @PutMapping | @DeleteMapping |
| Addl. Annotations | @RequestBody | @PathVariable | @PathVariable, @RequestBody | @PathVariable |
| MethodName | AddEntity(Obj obj) | findAll(), findById (int id) | updateEntity(int id, Obj obj) | deleteEnityt(int id) |

AssociateRepo implements

#

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

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

#spring.datasource.username=sa

#spring.datasource.password=

#spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

##

#spring.h2.console.enabled=true

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

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

spring.datasource.username=root

spring.datasource.password=root

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

spring.jpa.hibernate.ddl-auto= update

spring.jpa.show-sql=true

#spring.datasource.url=jdbc:postgresql://localhost:5432/tcs

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

#spring.datasource.username=postgres

#spring.datasource.password=postgres

#spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect