Spring Data --- It’s a JPA Implementation of Spring Boot. It simplifies the process of interacting to the database.

Database CRUD Operation

C – Create/Insert Operation

R – Read Operation (ReadAll, ReadById)

U – Update Operation

D – Delete Operation

Spring Web Services

SOAP based Web Service (Simple Object Access Protocol --

REST based Web Service ( Hyper Text Transfer Protocol --- get, post, put, delete )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Operation | Http Method | Requirements | Syntax | Return Type | Mapping Used |
| 1 | Create | POST | It needs a bean object in the Request body | Save(Bean obj) | Void or Object | @PostMapping(“/users”) |
| 2 | Read | GET | Either the primary key in the PathVariable or none | findAll(), findById(int id) | List<Bean> or Single Bean Object | @GetMapping |
| 3 | Update | PUT | Id in the PathVariable and Updated Object in Request Body | save(Bean obj) | Int or void | @PutMapping |
| 4 | Delete | DELETE | Id in the PathVariable | deleteById(int id) | Int or void | @DeleteMapping |

DAO – Data Access Object (Using this we will do the CRUD operation on the DB Table)

1. ReadAll(), ReadOne(int id)
2. Save(Object obj)
3. Update(int id, Object obj)
4. Delete(int id)

|  |  |  |
| --- | --- | --- |
| Sl No | Operation | Sample Service Code |
| 1 | Post (Create) | @PostMapping(“/users”)  public void saveUser(User user) {  userRepo.save(user);  } |
| 2 | Get (Read) | @GetMapping (“/users”)  public List<User> findAll() {  return userRepo.findAll();  }  @GetMapping(“/users/{id}”)  public User findById(@PathVariable int id){  return userRepo.findById(id).get();  } |
| 3 | Put (Update) | @PutMapping (“/users/{id}”)  Public void updateUser(@PathVariable int id, @RequestBody User user) {  userRepo.save(user);  } |
| 4 | Delete(Delete) | @DeleteMapping (“/users/{id}”)  Public void deleteUser(@PathVariable int id) {  userRepo.deleteById(id);  } |

Car Company (Audi) ---

Car is a type of Vehicle. Audi is a Specific type of Car.

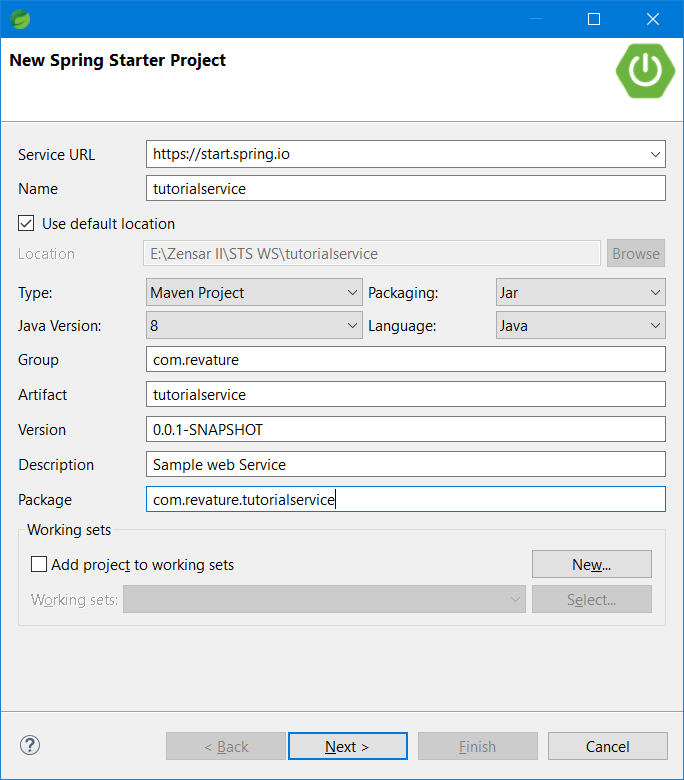
Car (Specification) – Audi Car (Implementation)

JPA (Specification) --- Hibernate (Implementation)

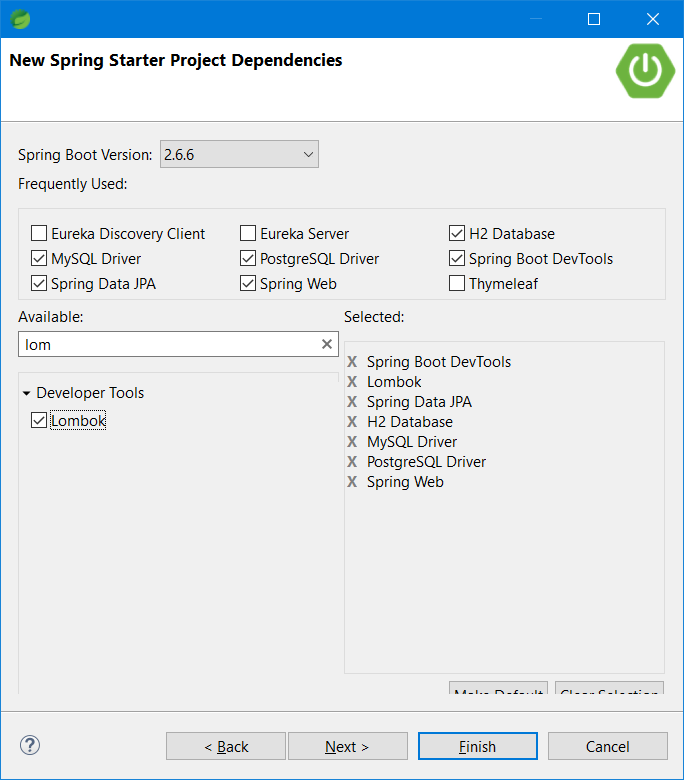
JDBC (Specification) --- MySQL Driver (Implementation) , Oracle Driver (Implementation)

<https://www.bezkoder.com/angular-11-spring-boot-crud/>

* Open STS
* Create a Spring Starter Project (File🡪 New🡪 Spring Starter Project)



* Add the following dependencies



Int id ;

String title;

String description;

String status;

**package** com.revature.tutorialservice;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** TutorialserviceApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(TutorialserviceApplication.**class**, args);

}

}

**package** com.revature.tutorialservice.model;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

**import** lombok.Data;

@Entity

@Table(name = "tutorial")

@Data

**public** **class** Tutorial {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id ;

**private** String title;

**private** String description;

**private** **boolean** published;

}

**package** com.revature.tutorialservice.repo;

**import** java.util.List;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** com.revature.tutorialservice.model.Tutorial;

**public** **interface** TutorialRepository **extends** JpaRepository<Tutorial, Integer> {

List<Tutorial> findByPublished(**boolean** published);

List<Tutorial> findByTitleContaining(String title);

}

**package** com.revature.tutorialservice.service;

**import** java.util.List;

**import** com.revature.tutorialservice.model.Tutorial;

**public** **interface** TutorialService {

**public** List<Tutorial> findAll();

**public** Tutorial findById(**int** id);

**public** List<Tutorial> findByPublished(**boolean** published);

**public** List<Tutorial> findByTitleContaining(String title);

**public** **void** addTutorial(Tutorial tutorial);

**public** **void** updateTutorial(**int** id, Tutorial tutorial);

**public** **void** deleteTutorial(**int** id);

}

**package** com.revature.tutorialservice.service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.revature.tutorialservice.model.Tutorial;

**import** com.revature.tutorialservice.repo.TutorialRepository;

@Service

**public** **class** TutorialServiceImpl **implements** TutorialService{

@Autowired

TutorialRepository tutorialRepository;

@Override

**public** List<Tutorial> findAll() {

// **TODO** Auto-generated method stub

**return** tutorialRepository.findAll();

}

@Override

**public** Tutorial findById(**int** id) {

// **TODO** Auto-generated method stub

**return** tutorialRepository.findById(id).get();

}

@Override

**public** List<Tutorial> findByPublished(**boolean** published) {

// **TODO** Auto-generated method stub

**return** tutorialRepository.findByPublished(published);

}

@Override

**public** List<Tutorial> findByTitleContaining(String title) {

// **TODO** Auto-generated method stub

**return** tutorialRepository.findByTitleContaining(title);

}

@Override

**public** **void** addTutorial(Tutorial tutorial) {

// **TODO** Auto-generated method stub

tutorialRepository.save(tutorial);

}

@Override

**public** **void** updateTutorial(**int** id, Tutorial tutorial) {

// **TODO** Auto-generated method stub

tutorialRepository.save(tutorial);

}

@Override

**public** **void** deleteTutorial(**int** id) {

// **TODO** Auto-generated method stub

tutorialRepository.deleteById(id);

}

}

Controller -🡪 Service -🡪 Repository -🡪 Database (post, put, delete)

Controller🡨 Service 🡨Repository 🡨 Database (Get operation)

**package** com.revature.tutorialservice.controller;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.web.bind.annotation.DeleteMapping;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.PostMapping;

**import** org.springframework.web.bind.annotation.PutMapping;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.revature.tutorialservice.model.Tutorial;

**import** com.revature.tutorialservice.service.TutorialService;

@RestController

@RequestMapping("/api")

**public** **class** TutorialController {

@Autowired

TutorialService tutorialService;

@GetMapping("/tutorials")

**public** List<Tutorial> findAll() {

// **TODO** Auto-generated method stub

**return** tutorialService.findAll();

}

@GetMapping("/tutorials/{id}")

**public** Tutorial findById(@PathVariable **int** id) {

// **TODO** Auto-generated method stub

**return** tutorialService.findById(id);

}

@GetMapping("/tutorials/published/{published}")

**public** List<Tutorial> findByPublished(@PathVariable **boolean** published) {

// **TODO** Auto-generated method stub

**return** tutorialService.findByPublished(published);

}

@GetMapping("/tutorials/{title}")

**public** List<Tutorial> findByTitleContaining(@PathVariable String title) {

// **TODO** Auto-generated method stub

**return** tutorialService.findByTitleContaining(title);

}

@PostMapping("/tutorials")

**public** **void** addTutorial(@RequestBody Tutorial tutorial) {

// **TODO** Auto-generated method stub

tutorialService.addTutorial(tutorial);

}

@PutMapping("/tutorials/{id}")

**public** **void** updateTutorial(@PathVariable **int** id, @RequestBody Tutorial tutorial) {

// **TODO** Auto-generated method stub

tutorialService.updateTutorial(id, tutorial);

}

@DeleteMapping("/tutorials/{id}")

**public** **void** deleteTutorial(**int** id) {

// **TODO** Auto-generated method stub

tutorialService.deleteTutorial(id);

}

}

server.port=8090

# h2 database configuration properties

#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

# MySQL Database Configuration

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

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

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

#Postgres database configuration

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

#spring.datasource.username= postgres

#spring.datasource.password= 123

#

#spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation= true

#spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect

#

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

## Hibernate ddl auto (create, create-drop, validate, update)

#spring.jpa.hibernate.ddl-auto= update

#

#logging.level.org.springframework=info

#logging.level.com.cognizant=debug

# Hibernate logs for displaying executed SQL, input and output

#logging.level.org.hibernate.SQL=trace

#logging.level.org.hibernate.type.descriptor.sql=trace

# Log pattern

#logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

Tutorial-frontend (Angular Application)

* Open VS Code
* Create a new angular app (ng new tutorial-frontent)
* ng g s services/tutorial
* ng g c components/add-tutorial
* ng g c components/tutorial-details
* ng g c components/tutorials-list
* ng g class model/tutorial

CORS – Cross Origin Resource Sharing

localhost:4200 --🡪 localhost:8090