Create A Spring Stater Project On Spring Tool Suit Named – Sporty\_Master

1.Add dependencies to Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.4</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>sporty\_master</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>sporty\_master</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</project>

2. Create Model

Package name - com.sporty.shoe.model

Create class – shoe

package com.sporty.shoe.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity

@Data

@AllArgsConstructor

@NoArgsConstructor

@Table(name = "Shoe")

public class Shoe {

@Id

@GeneratedValue

private int id;

private String name;

private String category;

private double price;

}

Create class – PurchaseReport

package com.sporty.shoe.model;

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity

@Data

@AllArgsConstructor

@NoArgsConstructor

public class PurchaseReport {

@Id

@GeneratedValue

private int id;

private String purchasedBy; // This can be extended to utilize one to one relation with User Table [Future Implemetations]

private String category;

private String orderList;

@Temporal(TemporalType.DATE)

private Date dop;

}

3. Create Repository

Create ShoesRepository Interface

**package** com.sporty.shoe.repository;

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

**import** com.sporty.shoe.model.Shoe;

**public** **interface** ShoesRepository **extends** JpaRepository<Shoe, Integer> {

}

Create PurchaseReportRepository Interface

package com.sporty.shoe.repository;

import java.util.Date;

import java.util.List;

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

import com.sporty.shoe.model.PurchaseReport;

public interface PurchaseReportRepository extends JpaRepository<PurchaseReport, Integer> {

public List<PurchaseReport> findByDop(Date dop);

public List<PurchaseReport> findByCategory(String category);

}

Create service

Create SportyShoesService Class

package com.sporty.shoe.service;

import java.util.Date;

import java.util.List;

import java.util.NoSuchElementException;

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

import org.springframework.dao.EmptyResultDataAccessException;

import org.springframework.stereotype.Service;

import com.sporty.shoe.exceptionHandler.BusinessException;

import com.sporty.shoe.model.PurchaseReport;

import com.sporty.shoe.model.Shoe;

import com.sporty.shoe.repository.PurchaseReportRepository;

import com.sporty.shoe.repository.ShoesRepository;

import lombok.NoArgsConstructor;

@Service

@NoArgsConstructor

public class SportyShoesService {

@Autowired

private ShoesRepository shoesRepo;

@Autowired

private PurchaseReportRepository prRepo;

public Shoe createShoe(Shoe shoe) throws BusinessException {

int id = shoe.getId();

Shoe oldShoe = null;

try {

oldShoe = shoesRepo.findById(id).get();

}catch(NoSuchElementException e) {

}

if(oldShoe!=null) throw new BusinessException("Shoe already exists with id: "+id);

return shoesRepo.save(shoe);

}

public Shoe getShoeById(int id) throws BusinessException {

Shoe shoe = null;

try {

if(id<=0) throw new BusinessException("Shoe Id can not be negative or zero");

shoe = shoesRepo.findById(id).get();

}catch(NoSuchElementException e) {

throw new BusinessException("Shoe not found with Id: "+id);

}

return shoe;

}

public Shoe updateShoe(Shoe shoe) {

return shoesRepo.save(shoe);

}

public void deleteShoeById(int id) throws BusinessException {

try {

shoesRepo.deleteById(id);

}catch(IllegalArgumentException e) {

throw new BusinessException("Invalid id: "+id);

}catch(EmptyResultDataAccessException e) {

throw new BusinessException("SHoe does not exist with id: "+id);

}

}

public List<Shoe> getAllShoes() {

return shoesRepo.findAll();

}

public PurchaseReport createPurchaseReport(PurchaseReport pr) throws BusinessException {

int id = pr.getId();

PurchaseReport oldPr = null;

try {

oldPr = prRepo.findById(id).get();

}catch(NoSuchElementException e) {

}

if(oldPr!=null) throw new BusinessException("Purchase report already exists with id: "+id);

return prRepo.save(pr);

}

public PurchaseReport getPurchaseReportById(int id) throws BusinessException {

PurchaseReport pr = null;

try {

if(id<=0) throw new BusinessException("Purchase Report Id can not be negative or zero");

pr = prRepo.findById(id).get();

}catch(NoSuchElementException e) {

throw new BusinessException("Purchase Report not found with Id: "+id);

}

return pr;

}

public PurchaseReport updatePurchaseReport(PurchaseReport pr) {

return prRepo.save(pr);

}

public void deletePurchaseReportById(int id) throws BusinessException {

try {

prRepo.deleteById(id);

}catch(IllegalArgumentException e) {

throw new BusinessException("Invalid id: "+id);

}catch(EmptyResultDataAccessException e) {

throw new BusinessException("Puchase Report does not exist with Id: "+id);

}

}

public List<PurchaseReport> getAllPurchaseReports() {

return prRepo.findAll();

}

public List<PurchaseReport> getAllPurchaseReportsByCategory(String category) {

return prRepo.findByCategory(category);

}

public List<PurchaseReport> getAllPurchaseReportsByDOP(Date dop) {

return prRepo.findByDop(dop);

}

}

Create Controller

Create CRUDController class

package com.sporty.shoe.controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.util.LinkedMultiValueMap;

import org.springframework.util.MultiValueMap;

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.RestController;

import com.sporty.shoe.exceptionHandler.BusinessException;

import com.sporty.shoe.model.PurchaseReport;

import com.sporty.shoe.model.Shoe;

import com.sporty.shoe.service.SportyShoesService;

@RestController

public class CRUDController {

@Autowired

private SportyShoesService service;

private MultiValueMap<String, String> errorMap;

@PostMapping("/admin/shoe")

public ResponseEntity<Shoe> createShoe(@RequestBody Shoe shoe) {

try {

return new ResponseEntity<>(service.createShoe(shoe), HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(null, errorMap, HttpStatus.BAD\_REQUEST);

}

}

@GetMapping("/admin/shoe/{id}")

public ResponseEntity<Shoe> getShoeById(@PathVariable int id) {

try {

return new ResponseEntity<>(service.getShoeById(id), HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(null, errorMap, HttpStatus.NOT\_FOUND);

}

}

@PutMapping("/admin/shoe")

public ResponseEntity<Shoe> updateShoe(@RequestBody Shoe shoe) {

return new ResponseEntity<>(service.updateShoe(shoe), HttpStatus.OK);

}

@GetMapping("/admin/shoe")

public List<Shoe> getAllshoescall() {

return service.getAllShoes();

}

@DeleteMapping("/admin/shoe/{id}")

public ResponseEntity<String> deleteShoeById(@PathVariable int id) {

try {

service.deleteShoeById(id);

return new ResponseEntity<>("Succesfully deleted shoe with id: " + id, HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(e.getMessage(), errorMap, HttpStatus.BAD\_REQUEST);

}

}

@PostMapping("/admin/purchaseReport")

public ResponseEntity<PurchaseReport> createPurchaseReport(@RequestBody PurchaseReport pr) {

try {

return new ResponseEntity<>(service.createPurchaseReport(pr), HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(null, errorMap, HttpStatus.BAD\_REQUEST);

}

}

@GetMapping("/admin/purchaseReport/id/{id}")

public ResponseEntity<PurchaseReport> getPurchaseReportById(@PathVariable int id) {

try {

return new ResponseEntity<>(service.getPurchaseReportById(id), HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(null, errorMap, HttpStatus.NOT\_FOUND);

}

}

@PutMapping("/admin/purchaseReport")

public ResponseEntity<PurchaseReport> updatePurchaseReport(@RequestBody PurchaseReport pr) {

return new ResponseEntity<>(service.updatePurchaseReport(pr), HttpStatus.OK);

}

@DeleteMapping("/admin/purchaseReport/{id}")

public ResponseEntity<String> deletePurchaseReportById(@PathVariable int id) {

try {

service.deletePurchaseReportById(id);

return new ResponseEntity<>("Succesfully deleted Purchase Report with id: " + id, HttpStatus.OK);

} catch (BusinessException e) {

errorMap = new LinkedMultiValueMap<>();

errorMap.add("errorMessage:", e.getMessage());

return new ResponseEntity<>(e.getMessage(), errorMap, HttpStatus.BAD\_REQUEST);

}

}

}

Create SearchController class

**package** com.sporty.shoe.controller;

**import** java.util.Date;

**import** java.util.List;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

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

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

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

**import** com.sporty.shoe.model.PurchaseReport;

**import** com.sporty.shoe.model.Shoe;

**import** com.sporty.shoe.service.SportyShoesService;

@RestController

**public** **class** SearchController {

@Autowired

**private** SportyShoesService service;

@GetMapping("/admin/shoe/all")

**public** ResponseEntity<List<Shoe>> getAllShoes(){

**return** **new** ResponseEntity<List<Shoe>>(service.getAllShoes(), HttpStatus.***OK***);

}

@GetMapping("/admin/purchaseReport/category/{category}")

**public** ResponseEntity<List<PurchaseReport>> getAllPurchaseReportsByCategory(@PathVariable String category){

**return** **new** ResponseEntity<List<PurchaseReport>>(service.getAllPurchaseReportsByCategory(category), HttpStatus.***OK***);

}

@GetMapping("/admin/purchaseReport/date/{dateInMs}")

**public** ResponseEntity<List<PurchaseReport>> getAllPurchaseReportsByDop(@PathVariable Long dateInMs){

Date dop = **new** Date(dateInMs);

**return** **new** ResponseEntity<List<PurchaseReport>>(service.getAllPurchaseReportsByDOP(dop), HttpStatus.***OK***);

}

@GetMapping("/admin/purchaseReport/all")

**public** ResponseEntity<List<PurchaseReport>> getAllPurchaseReport(){

**return** **new** ResponseEntity<List<PurchaseReport>>(service.getAllPurchaseReports(), HttpStatus.***OK***);

}

}

Create Exception

Create BusinessException

**package** com.sporty.shoe.exceptionHandler;

**public** **class** BusinessException **extends** Exception {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1008128726286682480L;

**public** BusinessException() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** BusinessException(String message) {

**super**(message);

// **TODO** Auto-generated constructor stub

}

}