Searching for a Specific User and Updating the User Information

package com.ecommerce;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@EnableJpaRepositories

@ComponentScan({"com.ecommerce.controller", "com.ecommerce.entity","com.ecommerce.repositry" })

@SpringBootApplication

public class SpringBootJpaHibernateDemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringBootJpaHibernateDemoApplication.class, args);

}

}

NoProductsWithSuchNameException2

**package** com.ecommerce.controller;

**public** **class** NoProductsWithSuchNameException **extends** Exception {

}

NoProductsWithSuchNameException

**package** com.ecommerce.controller;

**public** **class** NoProductsWithSuchNameException **extends** Exception {

}

ProductController.java

**package** com.ecommerce.controller;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Optional;

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

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

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

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

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

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

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

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

**import** com.ecommerce.entity.EProduct;

**import** com.ecommerce.exception.ProductSearchException;

**import** com.ecommerce.repositry.EProductRepositry;

@Controller

**public** **class** ProductController {

@Autowired

EProductRepositry eProductRepositry;

@GetMapping("/add-product")

**public** String showNewProductForm(Model model) {

EProduct new\_product = **new** EProduct();

model.addAttribute("product", new\_product);

**return** "new-product"; // go to new-product.jsp

}

@PostMapping("/add-product")

**public** String addProduct(@ModelAttribute("product") EProduct product) {

product.setDateAdded(**new** Date());

eProductRepositry.save(product);

**return** "new-product-added-success"; // go to new-product-added-success.jsp

}

// LIST ALL PRODUCTS

@GetMapping("/list-products")

**public** String listProduct(Model model) {

List<EProduct> products = eProductRepositry.findAll();

model.addAttribute("prodlist", products);

**return** "list-of-products"; // go to product-list.jsp

}

// DELETE A PRODUCT

@GetMapping("/delete-product")

**public** String deleteProduct(@RequestParam **long** id, Model model) {

Optional<EProduct> prodFromRepo = eProductRepositry.findById(id);

model.addAttribute("id", id);

**if** (prodFromRepo.isPresent()) {

eProductRepositry.deleteById(id);

} **else** {

**return** "delete-product-failed"; // go to delete-product-failed.jsp

}

;

**return** "delete-product-success"; // go to delete-product-success.jsp

}

// EDIT A PRODUCT: SHOW THE FORM

@GetMapping("/edit-product")

**public** String showEditProductForm(@RequestParam **long** id, Model model) {

Optional<EProduct> prodFromRepo = eProductRepositry.findById(id);

model.addAttribute("id", id);

**if** (prodFromRepo.isPresent()) {

EProduct product = prodFromRepo.get();

model.addAttribute("prod", product);

// Show the Edit Form

**return** "edit-product-form";

} **else** {

**return** "product-not-found";// product-not-found.jsp

}

}

// EDIT A PRODUCT: SAVE THE EDITED PRODUCT

@PostMapping("/edit-product")

**public** String saveEditedProduct(@ModelAttribute("prod") EProduct product) {

product.setDateAdded(**new** Date());

eProductRepositry.save(product);

**return** "edit-product-success"; // go to edit-product-success.jsp

}

// Advanced CRUD operations

// 5a. LIST ALL PRODUCTS that has a particular name

@GetMapping("/list-products-by-name")

**public** String listProductByName(@RequestParam String name, Model model) {

List<EProduct> products = eProductRepositry.findAllByName(name);

model.addAttribute("prodlist", products);

**return** "list-of-products"; // go to list-of-products.jsp

}

// 5b. LIST ALL PRODUCTS that has a particular price

@GetMapping("/list-products-by-price")

**public** String listProductByPrice(@RequestParam **float** price, Model model) {

List<EProduct> products = eProductRepositry.findAllByPrice(price);

model.addAttribute("prodlist", products);

**return** "list-of-products"; // go to list-of-products.jsp

}

//6.

@GetMapping("/list-products-by-name-like")

**public** String listProductByNameLike(@RequestParam String name, Model model) **throws** NoProductsWithSuchNameException, ProductSearchException {

**if**(name.length()<=1)

**throw** **new** ProductSearchException("Type atleast 3 chars to search");

List<EProduct> products = eProductRepositry.findAllByNameContaining(name);

// if(products.size()==0)

// throw new ProductSearchException("List is empty");

model.addAttribute("prodlist", products);

**return** "list-of-products"; // go to list-of-products.jsp

}

// Exception Handling

@ExceptionHandler

**public** ResponseEntity<Object> abc(Exception ex){

**return** **new** ResponseEntity<>("Sorry, something went wrong. Contact CC.", HttpStatus.***BAD\_REQUEST***);

}

@ExceptionHandler

**public** ResponseEntity<Object> abc1(NoProductsWithSuchNameException ex){

**return** **new** ResponseEntity<>("Sorry,"+ex.getMessage(), HttpStatus.***NOT\_FOUND***);

}

}

EProduct.java

package com.ecommerce.entity;

import java.math.BigDecimal;

import java.util.Date;

import jakarta.persistence.\*;

@Entity

@Table(name="eproduct")

public class EProduct {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="id")

private long ID;

private String name;

private BigDecimal price;

@Column(name="date\_added")

private Date dateAdded;

public EProduct() {

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

}

ProductSearchException.java

**package** com.ecommerce.exception;

**public** **class** ProductSearchException **extends** Exception {

**public** ProductSearchException(String message) {

**super**(message);

}

}

EProductRepositry.java

**p**ackage com.ecommerce.repositry;

**import** java.util.List;

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

**import** org.springframework.stereotype.Repository;

**import** com.ecommerce.entity.EProduct;

@Repository

**public** **interface** EProductRepositry **extends** JpaRepository<EProduct, Long>{

// Spring only wants to mention those CRUD methods

// that are not based on ID

List<EProduct> findAllByName(String name);

List<EProduct> findAllByPrice(**float** price);

//search by name like

List<EProduct> findAllByNameContaining(String name);

//

List<EProduct> findAllByPriceGreaterThan(**float** price);

List<EProduct> findAllByPriceGreaterThanAndPriceLessThan(**float** price1, **float** price2);

//Ordering

List<EProduct> findAllByPriceGreaterThanOrderByPrice(**float** price);

}

application.properties

server.port=80

#JSP files configuation

spring.mvc.view.prefix=/WEB-INF/views/

spring.mvc.view.suffix=.jsp

#Database MySQL

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

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

spring.datasource.username=root

spring.datasource.password=root

#spring.jpa.show-sql=true

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

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

delete-product-failed.jsp

Delete product with id=${id} Failed!

<br><br>

<a href=*"/"* > Back to HOME</a>

delete-product-success.jsp

Delete product with id=${id} Success!

<br><br>

<a href=*"/"* > Back to HOME</a>

edit-product-form.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<div align=*"center"*>

Edit Product ( id = ${prod.ID} ):<br>

<form:form action=*"edit-product"* method=*"POST"* modelAttribute=*"prod"*>

<form:input path=*"ID"* value=*"*${prod.ID}*"* hidden=*"true"*/>

<form:label path=*"name"* >Product name:</form:label>

<form:input path=*"name"* value=*"*${prod.name}*"* />

<br />

<form:label path=*"price"*>Product Price:</form:label>

<form:input path=*"price"* value=*"*${prod.price}*"*/>

<br />

<form:button>Edit and Save</form:button>

</form:form>

<br>

<a href=*"/"* > Cancel and Go Back to HOME</a>

edit-product-success.jsp

Product edited successfully!!

<br>

<a href=*"/"* > Back to HOME</a>

index.jsp

<a href=*"add-product"* >Add New product</a>

&nbsp;&nbsp;|&nbsp;&nbsp;

<a href=*"list-products"* >List products</a>

<br><br>

Delete a product:<br>

<form action=*"delete-product"* method=*"GET"*>

<input name=*"id"*>

<input type=*submit* value=*"Delete"*>

</form>

<br><br>

Edit a product:<br>

<form action=*"edit-product"* method=*"GET"*>

<input name=*"id"*>

<input type=*submit* value=*"Edit"*>

</form>

<br><br>

Search products by exact Name:<br>

<form action=*"list-products-by-name"* method=*"GET"*>

<input name=*"name"*>

<input type=*submit* value=*"SEARCH"*>

</form>

<br><br>

Search products by Name Like:<br>

<form action=*"list-products-by-name-like"* method=*"GET"*>

<input name=*"name"*>

<input type=*submit* value=*"SEARCH"*>

</form>

<br><br>

Search products by Price:<br>

<form action=*"list-products-by-price"* method=*"GET"*>

<input name=*"price"*>

<input type=*submit* value=*"SEARCH"*>

</form>

list-of-products.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<td align=*"center"*>

<p> List of Products </p>

<table border=*1*>

<tr>

<th>ID

<th>NAME

<th>PRICE

<th>DATE ADDED

<th>EDIT

<th>DELETE

</tr>

<c:forEach var=*"product"* items=*"*${prodlist}*"*>

<tr>

<td>${product.ID}

<td>${product.name}

<td>${product.price}

<td>${product.dateAdded}

<td><a href=*"edit-product?id=*${product.ID}*"* > EDIT </a>

<td><a href=*"delete-product?id=*${product.ID}*"* > DELETE </a>

</tr>

</c:forEach>

</table>

<br><br>

<a href=*"add-product"* >Add New product</a>

<br><br>

<a href=*"/"* > Back to HOME</a>

new-product.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<div align=*"center"*>

<form:form action=*"add-product"* method=*"post"* modelAttribute=*"product"*>

<form:label path=*"name"*>Product name:</form:label>

<form:input path=*"name"* />

<br />

<form:label path=*"price"*>Product Price:</form:label>

<form:input path=*"price"* />

<br />

<form:button>Add Product</form:button>

</form:form>

<br><br>

<a href=*"/"* > Back to HOME</a>

new-product-added-success.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

Product added successfully!!

<br>

Details of the product added:

Name: ${product.name}, Price: ${product.price}

<br><br>

<a href=*"/"* > Back to HOME</a>

product-not-fount.jsp

Product with id=${id} Not found!

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>3.2.1</version>

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

</parent>

<groupId>com.ecommerce</groupId>

<artifactId>spring-boot-jpa-hibernate-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-boot-jpa-hibernate-demo</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>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<!-- JSP support in Spring -->

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>