package com.example.controller;

import java.util.List;

import java.util.Optional;

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.example.model.Product;

import com.example.service.ProductRepository;

public class ProductController {

*@*Autowired

ProductRepository productrepository;

// retrieve all product from database

*@*GetMapping("all")

public List<Product> getAllProduct()

{

List<Product> product=(List<Product>) productrepository.findAll();

return product;

}

// insert new product into database

*@*PostMapping("add")

public Product addProduct(*@*RequestBody Product product)

{

return productrepository.save(product);

}

// get particular product by their ID

*@*GetMapping("product/{id}")

public Optional<Product> getProductId(*@*PathVariable int id)

{

return productrepository.findById(id);

}

// update existing product

*@*PutMapping("update/{id}")

public Product updateProduct(*@*RequestBody Product newProduct, *@*PathVariable int id)

{

return productrepository.findById(id).map(product -> {

product.setId(newProduct.getId());

product.setName(newProduct.getName());

product.setSeason(newProduct.getSeason());

product.setBrand(newProduct.getBrand());

product.setCategory(newProduct.getCategory());

product.setPrice(newProduct.getPrice());

product.setDiscount(newProduct.getDiscount());

product.setColor(newProduct.getColor());

return productrepository.save(product);

}).orElseGet(() -> {

newProduct.setId(id);

return productrepository.save(newProduct);

});

}

// delete particular product from database

*@*DeleteMapping("delete/{id}")

public void deleteProduct(*@*PathVariable int id)

{

productrepository.deleteById(id);

}

//search products

*@*GetMapping("search/{id}")

public Optional<Product> searchProduct(*@*PathVariable int id)

{

return productrepository.findById(id);

}

// search by season

*@*GetMapping("searchbyseason/{season}")

public List<Product> searchProductbyseason(*@*PathVariable String season)

{

List<Product> searchByseason = productrepository.findBySeason(season);

return searchByseason;

}

// search by brand

*@*GetMapping("searchbybrand/{brand}")

public List<Product> searchProductbybrand(*@*PathVariable String brand)

{

List<Product> searchByBrand = productrepository.findByBrand(brand);

return searchByBrand;

}

//search by category

*@*GetMapping("searchbycategory/{category}")

public List<Product> searchProductbycategory(*@*PathVariable String category)

{

List<Product> searchByCategory = productrepository.findByCategory(category);

return searchByCategory;

}

*@*GetMapping("searchbyprice/{price}")

public Optional<Product> searchProductbyprice(*@*PathVariable int price)

{

return productrepository.findById(price);

}

*@*GetMapping("searchbycolor/{color}")

public List<Product> searchProductbycolor(*@*PathVariable String color)

{

List<Product> searchBycolor = productrepository.findByColor(color);

return searchBycolor;

}

}

}