package com.kitchenstory.controller;

import java.text.DateFormat;

import java.util.Calendar;

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.CrossOrigin;

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.RequestBody;

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

import com.kitchenstory.entities.Order;

import com.kitchenstory.entities.Product;

import com.kitchenstory.services.OrderService;

import com.kitchenstory.services.ProductService;

@RestController

@CrossOrigin(origins = "\*")

public class UserController {

@Autowired

private OrderService service;

@Autowired

private ProductService pservice;

@PostMapping("/order")

public ResponseEntity<?> createOrder(@RequestBody Order order){

DateFormat df = DateFormat.getDateInstance();

Calendar cl = Calendar.getInstance();

String od = df.format(cl.getTime());

order.setOd(od);

this.service.saveOrder(order);

return ResponseEntity.status(HttpStatus.CREATED).build();

}

@GetMapping("/getOrders")

public ResponseEntity<?> getAllOrders(){

List<Order> orders = this.service.getOrders();

if(orders.isEmpty()) {

return ResponseEntity.status(HttpStatus.NOT\_FOUND).build();

}

else {

return ResponseEntity.ok(orders);

}

}

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

public ResponseEntity<?> deleteOrder(@PathVariable("id") String id){

this.service.deleteOrder(id);

return ResponseEntity.status(HttpStatus.OK).build();

}

@GetMapping("/getProduct/{name}")

public ResponseEntity<?> getProducts(@PathVariable("name") String name){

List<Product> products = this.pservice.getAllByName(name);

if(products.isEmpty()) {

return ResponseEntity.status(HttpStatus.NOT\_FOUND).build();

}else {

return ResponseEntity.ok(products);

}

}

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

public ResponseEntity<?> getOrder(@PathVariable("id") String id){

Order order = this.service.getOrder(id);

if(order==null) {

return ResponseEntity.status(HttpStatus.NOT\_FOUND).build();

}else {

return ResponseEntity.ok(order);

}

}

}