package com.vikram.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.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.RequestMapping;

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

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

import com.vikram.dto.CustomerDto;

import com.vikram.entity.Customer;

import com.vikram.entity.Engineer;

import com.vikram.entity.Manager;

import com.vikram.service.UserService;

@RestController

@CrossOrigin

@RequestMapping("/users")

public class UserController {

@Autowired

private UserService userService;

@GetMapping("/")

public ResponseEntity<String> apiTest(){

return new ResponseEntity<String>("Hi user", HttpStatus.OK);

}

@GetMapping("/zipcodes")

public List<String> getZipcodes(){

System.out.println(userService.getZipcodes());

return userService.getZipcodes();

}

@PostMapping("/customers/new")

public String createCustomer(@RequestBody CustomerDto customerDto) {

Customer customer = new Customer(customerDto);

return userService.addCustomer(customer);

}

@PostMapping("/engineers/new")

public String createEngineer(@RequestBody Engineer engineer) {

return userService.addEngineer(engineer);

}

@PostMapping("/managers/new")

public String createManager(@RequestBody Manager manager) {

return userService.addManager(manager);

}

@GetMapping("/customers")

public List<Customer> getAllCustomers(){

return userService.getAllCustomers();

}

@GetMapping("customers/{id}")

public Customer getCustomerById(@PathVariable String id) {

return userService.getCustomerById(Long.parseLong(id));

}

@GetMapping("customers/findByUserId/{id}")

public Customer findCustomerByUserId(@PathVariable String id) {

return userService.findCustomerByUserId(Long.parseLong(id));

}

@GetMapping("engineers/findByUserId/{id}")

public Engineer findEngineerByUserId(@PathVariable String id) {

return userService.findEngineerByUserId(Long.parseLong(id));

}

@GetMapping("managers/findByUserId/{id}")

public Manager findBYUserId(@PathVariable String id) {

return userService.findManagerByUserId(Long.parseLong(id));

}

@GetMapping("engineers/{id}")

public Engineer getEngineerById(@PathVariable String id) {

return userService.getEngineerById(Long.parseLong(id));

}

@GetMapping("managers/{id}")

public Manager getManagerById(@PathVariable String id) {

return userService.getManagerById(Long.parseLong(id));

}

@GetMapping("/engineers")

public List<Engineer> getAllEngineers(){

return userService.getAllEngineers();

}

@GetMapping("engineers/byZipcode/{zipcode}")

public List<Engineer> getEngineersByZipcode(@PathVariable String zipcode){

return userService.getEngineersByZipcode(zipcode);

}

@GetMapping("/managers")

public List<Manager> getAllManagers(){

return userService.getAllManagers();

}

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

public String deleteCustomer(@PathVariable String id) {

return userService.deleteCustomer(Long.parseLong(id));

}

}