package com.simplilearn.medicare.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.PutMapping;

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

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

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

import com.simplilearn.medicare.entity.Customer;

import com.simplilearn.medicare.service.CustomerService;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

@RequestMapping("/api/customer")

public class CustomerController {

@Autowired

private CustomerService service;

@PostMapping("")

public ResponseEntity<Object> saveCustomer(@RequestBody Customer cust){

Customer resp=service.saveCustomer(cust);

if(resp!=null)

return new ResponseEntity<>(resp,HttpStatus.CREATED);

else

return new ResponseEntity<>("Error while creating object",HttpStatus.INTERNAL\_SERVER\_ERROR);

}

@GetMapping("")

public List<Customer> getAll(){

return service.getAllCustomers();

}

@GetMapping("{id}")

public ResponseEntity<Object> getById(@PathVariable int id){

Customer resp=service.getCustById(id);

if(resp!=null)

return new ResponseEntity<>(resp,HttpStatus.OK);

else

return new ResponseEntity<>("no user available",HttpStatus.NOT\_FOUND);

}

@PutMapping("{id}")

public ResponseEntity<Object> updateCustomer(@RequestBody Customer c,@PathVariable int id)

{

Customer resp=service.updateCustomer(c, id);

if(resp!=null)

return new ResponseEntity<>(resp,HttpStatus.OK);

else

return new ResponseEntity<>("no user available to update",HttpStatus.NOT\_FOUND);

}

@DeleteMapping("{id}")

public ResponseEntity<Object> deleteCustomer(@PathVariable int id){

if(service.deleteCustomer(id))

return new ResponseEntity<>("user Deleted successfully",HttpStatus.OK);

else

return new ResponseEntity<>("no user available to delete",HttpStatus.NOT\_FOUND);

}

}