package com.hexa.apicodingchallenge.controller;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Collectors;

import org.modelmapper.ModelMapper;

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.hexa.apicodingchallenge.dto.TaskDto;

import com.hexa.apicodingchallenge.exception.ResourceNotFoundException;

import com.hexa.apicodingchallenge.model.Task;

import com.hexa.apicodingchallenge.service.TaskService;

import jakarta.validation.Valid;

@CrossOrigin(origins = "\*")

@RestController

@RequestMapping("/api/tasks")

public class TaskController {

@Autowired

private TaskService taskService;

@Autowired

private ModelMapper modelMapper;

@PostMapping("/add")

public ResponseEntity<TaskDto> addTask(@Valid @RequestBody TaskDto taskDto) {

Task taskModel = this.taskService.addTask(taskDto);

TaskDto taskResponse = this.modelMapper.map(taskModel, TaskDto.class);

return ResponseEntity.ok(taskResponse);

}

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

public ResponseEntity<TaskDto> getTaskById(@Valid @PathVariable(name = "id") int id) throws ResourceNotFoundException {

Task taskModel = this.taskService.getTaskById(id);

TaskDto taskResponse = this.modelMapper.map(taskModel, TaskDto.class);

return ResponseEntity.ok(taskResponse);

}

@GetMapping("/getall")

public ResponseEntity<List<TaskDto>> getAllTasks() {

List<Task> taskModelList = this.taskService.getAllTasks();

List<TaskDto> taskDtoList = new ArrayList();

for(Task t : taskModelList) {

TaskDto dt = this.modelMapper.map(t, TaskDto.class);

taskDtoList.add(dt);

}

return ResponseEntity.ok(taskDtoList);

}

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

public ResponseEntity<TaskDto> updateTask(@Valid @PathVariable(name = "id") int id, @Valid @RequestBody TaskDto taskDto) throws ResourceNotFoundException {

Task updatedTask = this.taskService.updateTask(id, taskDto);

TaskDto taskResponse = this.modelMapper.map(updatedTask, TaskDto.class);

return ResponseEntity.ok(taskResponse);

}

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

public ResponseEntity<TaskDto> deleteTaskById(@Valid @PathVariable(name = "id") int id) throws ResourceNotFoundException {

Task deletedTask = this.taskService.deleteTaskById(id);

TaskDto taskResponse = this.modelMapper.map(deletedTask, TaskDto.class);

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

}

}