1. Task Management System

1. Create an entity class **Task**

Create a **Task** entity class with the necessary fields. Annotate the class with **@Entity** and specify the primary key with **@Id**.

id(long)

description(String)

completed(Boolean)

2. Create a Repository class **TaskRepository**

Create a **TaskRepository** interface by extending **JpaRepository**. This interface will provide basic CRUD operations for the **Task** entity.

**3.** Create a Service Layer class **TaskService:**

Create a service class **TaskService** that will use the **TaskRepository** to perform business logic.

4. Create a **Controller** class **TaskController:**

Create a Simple Controller class **TaskController** to expose the functionality.

Create a REST controller to expose the functionality through HTTP endpoints.

5. Create **Application Properties:**

Ensure that your **application.properties** or **application.yml** file includes the necessary configurations for the MySQL database and Spring Data JPA.

**Answer:**

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Task {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String description;

private boolean completed;

// Constructors, getters, and setters

}

**TaskRepository**.**java**

import org.springframework.data.jpa.repository.JpaRepository;

public interface TaskRepository extends JpaRepository<Task, Long> {

// Additional custom queries can be added here if needed

}

**TaskService**.**java**

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class TaskService {

private final TaskRepository taskRepository;

@Autowired

public TaskService(TaskRepository taskRepository) {

this.taskRepository = taskRepository;

}

public List<Task> getAllTasks() {

return taskRepository.findAll();

}

public Optional<Task> getTaskById(Long taskId) {

return taskRepository.findById(taskId);

}

public Task saveTask(Task task) {

return taskRepository.save(task);

}

public void deleteTask(Long taskId) {

taskRepository.deleteById(taskId);

}

}

**TaskController**.**java**

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/tasks")

public class TaskController {

private final TaskService taskService;

@Autowired

public TaskController(TaskService taskService) {

this.taskService = taskService;

}

@GetMapping

public List<Task> getAllTasks() {

return taskService.getAllTasks();

}

@GetMapping("/{taskId}")

public ResponseEntity<Task> getTaskById(@PathVariable Long taskId) {

Optional<Task> task = taskService.getTaskById(taskId);

return task.map(ResponseEntity::ok).orElseGet(() -> ResponseEntity.notFound().build());

}

@PostMapping

public Task createTask(@RequestBody Task task) {

return taskService.saveTask(task);

}

@DeleteMapping("/{taskId}")

public ResponseEntity<Void> deleteTask(@PathVariable Long taskId) {

taskService.deleteTask(taskId);

return ResponseEntity.noContent().build();

}

}

spring.datasource.url=””

spring.datasource.driverClassName=””

spring.datasource.username=””

spring.datasource.password=””

spring.jpa.database-platform=””