**Spring Data JPA - Quick Example**

## Step 1: application.properties

### ➤ For H2 (in-memory DB)

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

spring.jpa.hibernate.ddl-auto=update

### ➤ For MySQL

spring.datasource.url=jdbc:mysql://localhost:3306/springdb

spring.datasource.username=root

spring.datasource.password=yourpassword

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

## Step 2: Create Entity

import jakarta.persistence.\*;

@Entity

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String department;

}

## Step 3: Create Repository

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

public interface StudentRepository extends JpaRepository<Student, Long> {

}

## Step 4: Create Controller

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

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

import java.util.List;

@RestController

@RequestMapping("/students")

public class StudentController {

@Autowired

private StudentRepository studentRepo;

@PostMapping

public Student create(@RequestBody Student student) {

return studentRepo.save(student);

}

@GetMapping

public List<Student> getAll() {

return studentRepo.findAll();

}

@GetMapping("/{id}")

public Student getById(@PathVariable Long id) {

return studentRepo.findById(id).orElse(null);

}

@PutMapping("/{id}")

public Student update(@PathVariable Long id, @RequestBody Student student) {

student.setId(id);

return studentRepo.save(student);

}

@DeleteMapping("/{id}")

public void delete(@PathVariable Long id) {

studentRepo.deleteById(id);

}

}

## Test API with Postman or Swagger

POST /students → Add student

GET /students → Get all

GET /students/{id} → Get by ID

PUT /students/{id} → Update

DELETE /students/{id} → Delete