**Spring Data JPA - Quick Example**

**Code:**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>springdatajpa-demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.3</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version> <!-- Change to 21 if using Java 21 -->

</properties>

<dependencies>

<!-- Spring Boot Starter Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- MySQL Connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Starter Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Boot Starter -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Downgrade maven-resources-plugin to fix Eclipse error -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- Set compiler version -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Student.java**

package com.example.springdatajpa.entity;

import jakarta.persistence.\*;

@Entity

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String department;

// Getters and setters

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

**StudentRepository.java**

package com.example.springdatajpa.repository;

import com.example.springdatajpa.entity.Student;

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

public interface StudentRepository extends JpaRepository<Student, Long> {

// You can define query methods here if needed

}

**StudentController.java**

package com.example.springdatajpa.controller;

import com.example.springdatajpa.entity.Student;

import com.example.springdatajpa.repository.StudentRepository;

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

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

import java.util.Optional;

@RestController

@RequestMapping("/students")

public class StudentController {

@Autowired

private StudentRepository repo;

@PostMapping

public Student create(@RequestBody Student s) {

return repo.save(s);

}

@GetMapping("/{id}")

public Optional<Student> get(@PathVariable Long id) {

return repo.findById(id);

}

@DeleteMapping("/{id}")

public void delete(@PathVariable Long id) {

repo.deleteById(id);

}

}

**SpringdatajpaDemoApplication.java**

package com.example.springdatajpa;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringdatajpaDemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringdatajpaDemoApplication.class, args);

}

}

**application.properties**

# MySQL Configuration

spring.datasource.url=jdbc:mysql://localhost:3306/student\_db

spring.datasource.username=root

spring.datasource.password=your\_password

# JPA Configuration

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

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

**Output:**





