**SPRING DATA JPA WITH SPRING BOOT, HIBERNATE**

**1.Spring Data JPA – Quick Example**

**SpringDataJpaExample\src\main\java\com\example\springdatajpa\controller\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.List;

@RestController

@RequestMapping("/students")

public class StudentController {

    @Autowired

    private StudentRepository studentRepository;

    @GetMapping

    public List<Student> getAllStudents() {

        return studentRepository.findAll();

    }

    @PostMapping

    public Student createStudent(@RequestBody Student student) {

        return studentRepository.save(student);

    }

}

**SpringDataJpaExample\src\main\java\com\example\springdatajpa\entity\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 email;

    //Required getters and setters (for Spring/Jackson to work)

    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 getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

}

**SpringDataJpaExample\src\main\java\com\example\springdatajpa\repository\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> {

}

**SpringDataJpaExample\src\main\java\com\example\springdatajpa\SpringDataJpaApplication.java**

package com.example.springdatajpa;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringDataJpaApplication {

    public static void main(String[] args) {

        SpringApplication.run(SpringDataJpaApplication.class, args);

    }

}

**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

                             http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>springdatajpa</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>jar</packaging>

    <name>Spring Data JPA Example</name>

    <description>Simple Spring Boot project using Spring Data JPA</description>

    <parent>

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

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

        <version>3.2.5</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <dependency>

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

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

        </dependency>

        <dependency>

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

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

        </dependency>

        <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>

            <scope>runtime</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

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

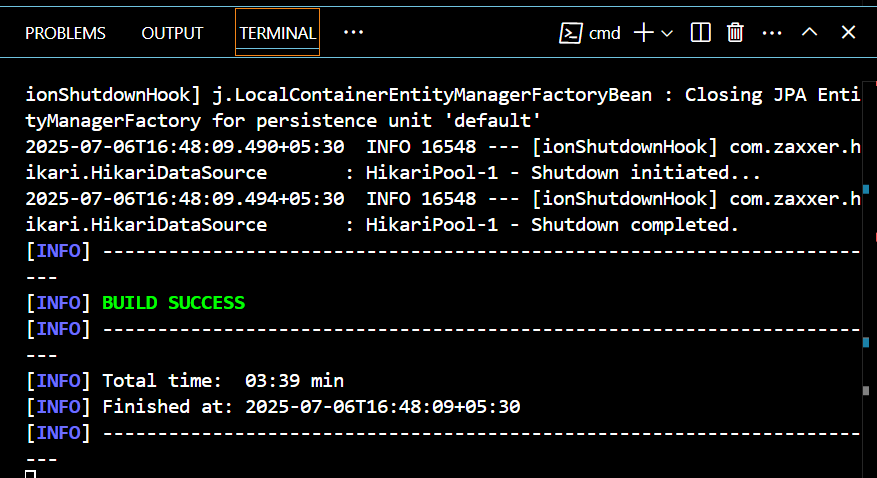
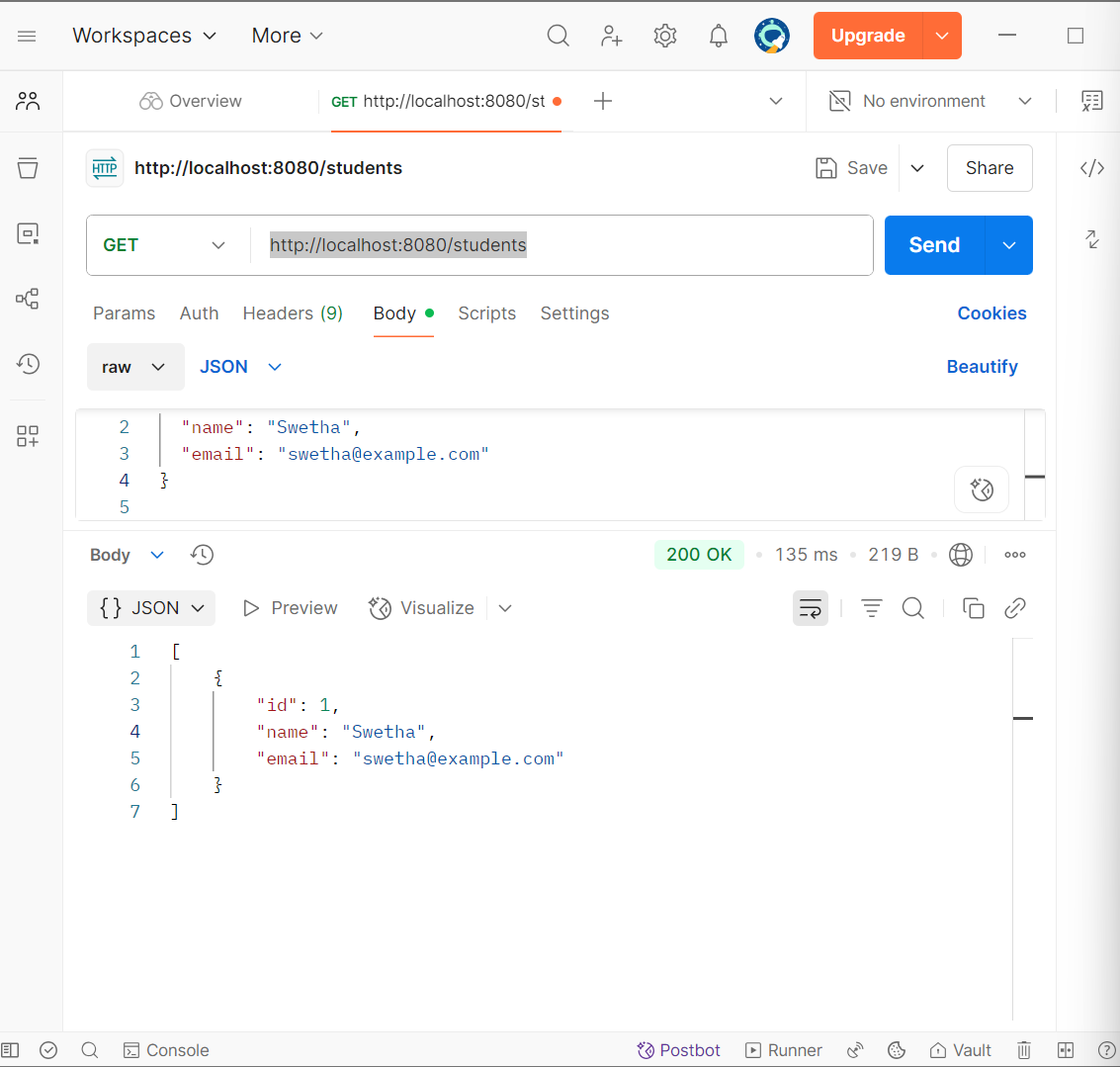
                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**OUTPUT :**

**2.Difference between JPA, Hibernate and Spring Data JPA**

**JPA (Java Persistence API)**

* **Type:** Specification (JSR 338)
* **Role:** Defines a set of interfaces and rules for object-relational mapping (ORM) in Java.
* **Implementation:** It does not provide an implementation. Instead, it is implemented by frameworks like Hibernate, EclipseLink, etc.
* **Purpose:** Allows Java objects to be stored and retrieved from a relational database in a standard way.
* **Example Interfaces:** EntityManager, Query, CriteriaBuilder

**Hibernate**

* **Type:** ORM Framework and JPA implementation
* **Implements:** JPA interfaces, but also provides additional features beyond JPA (like caching, lazy loading, HQL).
* **Purpose:** Maps Java classes to database tables and provides methods to perform CRUD operations.
* **Usage:** Can be used standalone or as the underlying JPA provider in Spring applications.

**Spring Data JPA**

* **Type:** Abstraction on top of JPA (and Hibernate by default)
* **Built On:** JPA + Hibernate + Spring Framework
* **Purpose:** Minimizes boilerplate code for data access layers.
* **Key Features:**
  + Repository interfaces (JpaRepository, CrudRepository)
  + Auto query generation (e.g., findByName)
  + Paging, sorting, and custom queries
* **Advantage:** Developers can perform complex DB operations **without writing any SQL or HQL**.