WEEK-3

Spring Data JPA

HandsOn 4

**Difference between JPA, Hibernate and Spring Data JPA**   
  
Java Persistence API (JPA)

* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementation of JPA

Hibernate

* ORM Tool that implements JPA

Spring Data JPA

* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transactions

Hibernate

   /\* Method to CREATE an employee in the database \*/

   public Integer addEmployee(Employee employee){

      Session session = factory.openSession();

      Transaction tx = null;

      Integer employeeID = null;

      try {

         tx = session.beginTransaction();

         employeeID = (Integer) session.save(employee);

         tx.commit();

      } catch (HibernateException e) {

         if (tx != null) tx.rollback();

         e.printStackTrace();

      } finally {

         session.close();

      }

      return employeeID;

   }

Spring Data JPA  
EmployeeRespository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java

@Autowire

  private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

  employeeRepository.save(employee); }

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

* **JPA (Java Persistence API):**

1. JPA is a **Java specification (JSR 338)** that defines how Java objects interact with relational databases.
2. It provides **standard interfaces and annotations** for object-relational mapping (ORM).
3. JPA **does not provide a concrete implementation**; it requires a provider like Hibernate or EclipseLink.
4. It offers APIs like EntityManager, @Entity, @Table, etc., to perform database operations.
5. JPA ensures **vendor independence,** allowing developers to switch between different ORM tools easily.

* **Hibernate:**

1. Hibernate is a popular ORM framework that provides a concrete implementation of the JPA specification.
2. It includes additional features beyond JPA, like caching, lazy loading, and batch processing.
3. Developers need to manage SessionFactory, transactions, and sessions manually (unless integrated with Spring).
4. Hibernate uses HQL (Hibernate Query Language), which is similar to JPQL.
5. Hibernate can be used both with and without JPA.

* **Spring Data JPA:**

1. Spring Data JPA is a **part of the Spring ecosystem** that simplifies data access using JPA.
2. It is **not a JPA implementation** but a **wrapper/abstraction over JPA** implementations like Hibernate.
3. It **eliminates boilerplate code** by providing ready-made interfaces like JpaRepository.
4. It supports **derived query methods**, custom queries using @Query, and automatic transaction management.
5. It is highly **declarative and reduces manual configuration,** promoting faster development with less code.