**Spring-data-jpa-handson**

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

**JPA (Java Persistence API):**

Official **JSR 338 specifications** for persisting, reading, and managing Java objects into relational databases. It is an interface — no implementation by itself.

**Hibernate:**

**ORM (Object-Relational Mapping) framework** and one of the popular implementations of the **JPA** specification. It internally manages database connections, sessions, transactions, and mapping of Java objects to database tables.

**Spring Data JPA:**

An abstraction layer on top of **JPA/Hibernate** that reduces boilerplate code and makes database interactions easier. It provides built-in CRUD operations, pagination, sorting, and custom query method naming without writing implementation code.

**Code Comparison:**

**Hibernate Code (Manual Session/Transaction Handling)**

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;

}

**Drawbacks**:

* You have to manually handle **Session**, **Transaction**, **exception handling**, and **closing resources**.

**Spring Data JPA Code (Simplified)**

**EmployeeRepository.java**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**Advantages**:

* No need to manually handle Session or Transaction.
* Built-in CRUD methods (save(), findAll(), deleteById(), etc.) via JpaRepository.
* Spring handles transaction management and resource cleanup automatically.

| **Hibernate (Manual ORM)** | **Spring Data JPA (Abstraction Layer)** |
| --- | --- |
| Manual session and transaction handling | Spring handles sessions and transactions automatically |
| More boilerplate code | Minimal code — just extend JpaRepository |
| Need to write HQL/SQL queries manually | Can use query method naming conventions or @Query annotation |
| Provides full control over ORM behavior | Less control but faster and cleaner implementation for common use cases |