**Hands-On 4: Difference between JPA, Hibernate, and Spring Data JPA**

**Java Persistence API (JPA)**

* JPA is a **Java Specification (JSR 338)** for managing relational data in Java applications.
* It defines the way to persist, read, and manage data from Java objects to relational databases.
* JPA **does not provide a concrete implementation**, only interfaces and annotations.
* Hibernate is a popular implementation of JPA.

**Hibernate**

* Hibernate is an **Object-Relational Mapping (ORM)** tool that implements the JPA specification.
* It provides the actual implementation of the JPA interfaces.
* Developers can use Hibernate directly using its native API.

**Spring Data JPA**

* Spring Data JPA **does not implement JPA** itself.
* It provides a **higher-level abstraction** over JPA implementations like Hibernate.
* It significantly **reduces boilerplate code** by offering built-in repository interfaces.
* It also **manages transactions** and integrates smoothly with Spring's infrastructure.

**Code Comparison**

**Hibernate Example:**

/\* 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 Example:**

**EmployeeRepository.java**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java**

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

**Summary**

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification | ORM Tool | Spring abstraction layer |
| Implements JPA? | No | Yes | No |
| Boilerplate Reduction | No | Partial | Yes |
| Transaction Handling | Manual or via implementation | Manual or via implementation | Handled by Spring (with @Transactional) |
| Ease of Use | Medium | Medium | High |