**Hands on 4**

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

Here's a clear and structured comparison for **Hands-on 4: Difference between JPA, Hibernate, and Spring Data JPA**, based on your explanation and code:

**🔹 Java Persistence API (JPA)**

* **Type:** Specification (JSR 338)
* **Purpose:** Defines a standard for ORM (Object Relational Mapping) in Java
* **Key Point:** Only provides interfaces and annotations—**no implementation**
* **Examples:**
  + @Entity, @Table, @Id, EntityManager
* **Needs a Provider:** Like **Hibernate**, **EclipseLink**, or **OpenJPA**

**🔹 Hibernate**

* **Type:** ORM Tool / JPA Implementation
* **Purpose:** Actual framework that maps Java objects to database tables
* **Implements:** JPA specification
* **Can work with or without JPA**
* **Needs manual setup:** sessions, transactions, exception handling

**Example (Using Hibernate Directly):**

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**

* **Type:** Spring abstraction layer over JPA (not a JPA provider)
* **Purpose:** Simplifies JPA-based data access using repositories
* **Built on:** JPA and a provider like Hibernate
* **Reduces boilerplate:** No need to write DAO or transaction code
* **Provides:** CrudRepository, JpaRepository, query derivation, paging/sorting

**Example (Using Spring Data JPA):**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**🔄 Summary Table**

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification | Framework (JPA Provider) | Spring Abstraction Layer |
| Implementation | No | Yes | No (uses Hibernate or others) |
| Boilerplate Code | Some | More | Minimal |
| Transaction Handling | Manual | Manual | Automatic with @Transactional |
| Repository Support | No | No | Yes (built-in with Spring support) |