HANDS ON 4

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

**DIFFERENCE:**

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification (JSR 338) | ORM Implementation | Framework/Abstraction Layer |
| Implementation | No | Yes (implements JPA) | No (uses JPA/Hibernate under the hood) |
| Boilerplate Code | Medium | High | Very Low |
| Transaction Management | Manual via EntityManager | Manual via SessionFactory | Automatic via @Transactional |
| Focus | Interface/API | ORM logic and mapping | Repository abstraction and CRUD out-of-the-box |

**HIBERNATE EXAMPLE:**

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;

}

**SPRINGDATAJPA EXAMPLE:**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}