**Difference between JPA, Hibernate and Spring Data JPA  
Week 3 Mandatory HandsOn**

**🔹 Java Persistence API (JPA)**

* **What it is**: A specification (JSR 338) that defines how Java objects should be persisted to relational databases.
* **Key Point**: **Not an implementation** — it defines interfaces and annotations but no actual working code.
* **Usage**: Needs an implementation like **Hibernate**, **EclipseLink**, etc.

**🔹 Hibernate**

* **What it is**: A popular **ORM (Object Relational Mapping)** framework that **implements JPA**.
* **Role**: Provides the actual code to persist, read, update, and delete Java objects from the database.
* **Direct Hibernate API** can also be used without JPA.

**🔹 Spring Data JPA**

* **What it is**: A part of the Spring ecosystem that provides an **abstraction layer over JPA**.
* **Key Features**:
  + Reduces **boilerplate code**.
  + Adds features like **automatic query generation**, **pagination**, **sorting**, etc.
  + Still needs a JPA provider like Hibernate under the hood.
  + Manages **transactions and repository implementation** automatically.

**🔸 Code Comparison**

**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;

}

* **Manual session and transaction handling**
* More boilerplate code

**Using Spring Data JPA:**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

// No implementation needed

}

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

* **Repository pattern** is auto-implemented.
* **Spring manages** transactions and sessions.
* Clean, declarative, and concise.

**Summary Table**

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification | Framework / Implementation | Abstraction / Module |
| Boilerplate Code | Medium | High | Low |
| Implements JPA? | No | Yes | No (depends on JPA provider) |
| Manages Transactions | No | Manually | Yes (Spring manages it) |
| Provided By | Java EE | Hibernate.org | Spring Framework |

Submitted By:  
Name: Shreyansh Kumar  
Superset Id: 6422403