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

# 1. Java Persistence API (JPA)

- JPA is a specification (JSR 338) provided by Java for Object-Relational Mapping (ORM).  
- It provides a standard way to map Java objects to relational database tables.  
- Does not have its own implementation — it's just a set of interfaces and annotations.  
- Popular implementations: Hibernate, EclipseLink, OpenJPA.

# 2. Hibernate

- Hibernate is the most popular implementation of JPA.  
- It is a full-featured ORM tool that provides:  
 • Lazy/eager loading  
 • Caching  
 • Automatic schema generation  
 • HQL (Hibernate Query Language)  
- Handles low-level database operations internally using JDBC.

# 3. Spring Data JPA

- Framework built by Spring to work on top of JPA.  
- Adds abstraction and automation to reduce boilerplate code.  
- You write interfaces, and Spring Data provides the implementation automatically!  
- Integrates with Spring Boot, Spring MVC, and Spring Transaction Management.

## Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA | Notes |
| Type | Specification | Implementation of JPA | Abstraction over JPA + Spring |  |
| Boilerplate Code | Medium | High | Very Low |  |
| Requires Configuration | Yes | Yes | Minimal with Spring Boot |  |
| Query Language | JPQL | HQL, JPQL | Derived queries, JPQL |  |
| Transaction Management | No | Partial | Yes (@Transactional) |  |

## Code Comparison

### Using Hibernate (Manual)

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;  
}

### Using Spring Data JPA

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);  
 }  
}

## Summary

|  |  |
| --- | --- |
| Tool | Role |
| JPA | Contract/specification for ORM (no implementation) |
| Hibernate | Concrete implementation of JPA |
| Spring Data JPA | Abstraction layer to simplify JPA/Hibernate use |

## Reference Links

- https://dzone.com/articles/what-is-the-difference-between-hibernate-and-sprin-1

- https://www.javaworld.com/article/3379043/what-is-jpa-introduction-to-the-java-persistence-api.html