Difference Between JPA, Hibernate, and Spring Data JPA

# Java Persistence API (JPA)

JPA is a Java specification (JSR 338) for object-relational mapping (ORM). It defines how Java objects interact with relational databases but does not provide implementation. You need a provider like Hibernate to use JPA in an application.

# Hibernate

Hibernate is a powerful ORM framework that implements the JPA specification. It allows developers to map Java classes to database tables and provides features such as lazy loading, caching, and HQL (Hibernate Query Language).

# Spring Data JPA

Spring Data JPA is a part of the Spring Framework that simplifies JPA-based data access. It abstracts boilerplate code by providing automatic implementation of repository interfaces. It still relies on a JPA provider like Hibernate internally.

# Comparison Table

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA |
| Type | Specification | Framework | Spring Module |
| Implementation | No | Yes | No (uses Hibernate) |
| Boilerplate Code | Medium | High | Low |
| Query Language | JPQL | JPQL + HQL | JPQL |
| Transaction Management | Manual or container-managed | Manual or Spring | Spring-managed |
| Repository Abstraction | No | No | Yes |

# Example using JPA

EntityManagerFactory emf = Persistence.createEntityManagerFactory("my-unit");  
EntityManager em = emf.createEntityManager();  
EntityTransaction tx = em.getTransaction();  
  
tx.begin();  
Employee emp = new Employee();  
emp.setName("John");  
emp.setSalary(50000);  
em.persist(emp);  
tx.commit();  
  
em.close();  
emf.close();

# Example using Hibernate

SessionFactory factory = new Configuration().configure().buildSessionFactory();  
Session session = factory.openSession();  
Transaction tx = session.beginTransaction();  
  
Employee emp = new Employee();  
emp.setName("Jane");  
emp.setSalary(60000);  
session.save(emp);  
  
tx.commit();  
session.close();  
factory.close();

# Example using Spring Data JPA

EmployeeRepository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}

EmployeeService.java

@Service  
public class EmployeeService {  
 @Autowired  
 private EmployeeRepository repository;  
  
 @Transactional  
 public void addEmployee(String name, double salary) {  
 Employee emp = new Employee();  
 emp.setName(name);  
 emp.setSalary(salary);  
 repository.save(emp);  
 }  
}