# Hands-on 4: Difference between JPA, Hibernate, and Spring Data JPA

## Java Persistence API (JPA)

- JPA is a JSR 338 specification for persisting, reading, and managing data from Java objects.

- It provides a standard API for ORM (Object-Relational Mapping) in Java.

- JPA itself does not include any concrete implementation, it is just a specification.

- Frameworks like Hibernate, EclipseLink, OpenJPA are implementations of the JPA specification.

## Hibernate

- Hibernate is an ORM (Object-Relational Mapping) tool that implements the JPA specification.

- It provides additional features beyond the JPA standard, such as caching, better performance optimizations, and custom query support.

- Developers interact directly with Hibernate’s Session and Transaction APIs if not using JPA.

## Spring Data JPA

- Spring Data JPA is a Spring-based abstraction layer on top of JPA.

- It does not provide its own JPA implementation, but instead works with existing implementations like Hibernate.

- Spring Data JPA greatly reduces boilerplate code, allowing you to focus on defining repository interfaces and query methods.

- It also provides transaction management, paging, and auditing out-of-the-box.

# Code Comparison: Hibernate vs Spring Data JPA

Below is an example of how to create and save an Employee entity using Hibernate and Spring Data JPA.

### Using Hibernate

/\* Method to CREATE an employee in the database \*/  
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

@Autowired  
private EmployeeRepository employeeRepository;  
  
@Transactional  
public void addEmployee(Employee employee) {  
 employeeRepository.save(employee);  
}

# Summary

- JPA: Specification only, no implementation.

- Hibernate: Implementation of JPA with additional features, requires manual session/transaction handling.

- Spring Data JPA: Abstraction over JPA and its implementation, simplifies CRUD and transaction management.