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

# Step-by-Step Explanation

## 1. JPA (Java Persistence API)

- JPA is just a specification (JSR 338). It provides a set of interfaces and annotations to manage relational data in Java applications.  
- It does not contain any actual implementation.  
- Examples of JPA providers: Hibernate, EclipseLink, OpenJPA

## 2. Hibernate

- Hibernate is an ORM (Object Relational Mapping) tool and implementation of JPA.  
- Provides more functionality than JPA (e.g., caching, HQL).  
- Can work standalone or as a JPA provider.

## 3. Spring Data JPA

- Spring Data JPA is a Spring framework module that acts as an abstraction over JPA implementations (like Hibernate).  
- Reduces boilerplate code.  
- Uses interfaces to perform CRUD operations automatically.  
- Handles transactions using Spring.

# Step-by-Step Code Comparison

## Hibernate – Manual Approach

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;  
}

What’s manual here?  
- Opening session  
- Beginning and committing transaction  
- Handling exceptions  
- Closing session

## Spring Data JPA – Simplified Approach

1. 1. Repository Interface

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

1. 2. Service Class

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

What’s simplified here?  
- No need to manually manage session or transactions  
- Just call save() method

# Summary Table

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA |
| Type | Specification | Implementation of JPA | Abstraction layer over JPA |
| Implementation | No | Yes | Uses Hibernate or any JPA implementation |
| Boilerplate Code | High | Moderate | Very Low |
| Transactions | Manual | Semi-Automated | Automatically handled by Spring |
| Repository Support | No | Limited | Powerful with method name conventions |
| Setup Effort | Moderate | More | Very Simple with Spring Boot |