[Week 3] [Spring Data JPA] HandsOn

# Spring Data JPA - Quick Example

Spring Data JPA simplifies JPA development by providing ready-to-use repositories and eliminating boilerplate code.  
  
Example Flow:  
  
1. Entity Class:  
@Entity  
public class Employee {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private Long id;  
  
 private String name;  
 private Double salary;  
  
}  
  
2. Repository Interface:  
public interface EmployeeRepository extends JpaRepository<Employee, Long> {  
 List<Employee> findByName(String name);  
}  
  
3. Service   
@Autowired  
private EmployeeRepository employeeRepository;  
  
public void demo() {  
   
 Employee emp = new Employee();  
 emp.setName("John");  
 emp.setSalary(5000.0);  
 employeeRepository.save(emp);  
  
   
 List<Employee> employees = employeeRepository.findByName("John");  
 employees.forEach(e -> System.out.println(e.getName()));  
}  
  
Key Benefit:  
- No need to write SQL or HQL for common operations.  
- Powerful query methods by just naming conventions (e.g. findByName).

# Difference between JPA, Hibernate and Spring Data JPA

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| Type | Specification (Standard) | Implementation of JPA | Abstraction & Tools over JPA |
| Who Defines It? | Java EE / Jakarta EE | Hibernate team | Spring Framework |
| Purpose | Standard API for ORM in Java | ORM library implementing JPA | Simplify JPA/Hibernate usage in Spring apps |
| SQL Writing | Developer writes JPQL/SQL queries | Developer writes JPQL/SQL queries | Often avoids manual query writing using derived query methods |
| Boilerplate Code | Must write EntityManager handling | Uses Session or EntityManager, with extra Hibernate features | Eliminates much boilerplate with ready-made repositories |
| Portability | Works with any JPA provider | Hibernate-specific | Works with any JPA provider, but often uses Hibernate |

Summary:  
- JPA is just a specification (no implementation code).  
- Hibernate implements JPA plus many extra features.  
- Spring Data JPA builds on top of JPA/Hibernate to:  
 - auto-generate repository implementations  
 - reduce boilerplate  
 - provide derived queries like findByName