Week 3-Spring Data JPA with Spring Boot, Hibernate

2. spring-data-jpa-handson

**Difference between JPA, Hibernate, and Spring Data JPA**

| **Feature** | **JPA (Java Persistence API)** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| **Type** | Specification (interface) | Implementation (framework) | Framework built on top of JPA |
| **Purpose** | Defines the standard for ORM in Java | Provides actual ORM functionality | Simplifies JPA by reducing boilerplate code |
| **Ownership** | Provided by Java EE / Jakarta EE | Open-source by RedHat | Part of Spring ecosystem |
| **Implementation Needed** | Yes – needs provider (like Hibernate) | No – it's an implementation | No – wraps around JPA and ORM providers like Hibernate |
| **Annotations** | Defines annotations like @Entity, @Id, @OneToMany | Uses JPA + extra Hibernate-specific annotations | Uses JPA annotations + Spring-specific repository APIs |
| **Boilerplate Code** | Requires writing EntityManager logic manually | Same as JPA, but offers some automation | Eliminates most boilerplate with JpaRepository, etc. |
| **Query Support** | Supports JPQL (Java Persistence Query Language) | Supports JPQL + native SQL + Criteria API | Supports JPQL, native SQL + method-name queries |
| **Use Case** | When you want a standard approach across providers | When you want more advanced ORM features | When you want easy and quick Spring-based data access |