

Difference between JPA, Hibernate and Spring Data JPA

Feature	JPA (Java Persistence API)	Hibernate	Spring Data JPA
Type	Specification (Interface/Standard)	Implementation of JPA (and more)	Abstraction over JPA/Hibernate
What it does	Defines how Java objects map to database tables	Provides concrete implementation of JPA + features	Simplifies data access layer using repository model
Provider	Part of Java EE / Jakarta EE	Developed by Red Hat	Part of the Spring ecosystem
Examples	@Entity, @Table, EntityManager	Session, CriteriaBuilder, HQL, Caching	CrudRepository, JpaRepository, @Query
Boilerplate	Requires a lot of boilerplate code	Slightly less, but still requires DAO layer	Minimal code, auto-implemented queries
Learning Curve	Moderate	Higher (due to many advanced features)	Easy (Spring-style)
Query Language	JPQL	JPQL + HQL + native SQL	JPQL + method naming + @Query
Usage	Just defines API	Can be used standalone or via Spring Data JPA	Requires JPA provider (like Hibernate) underneath

Summary:

- **JPA:** Standard/specification (like a rulebook).
- **Hibernate:** A powerful ORM framework and a JPA provider.
- **Spring Data JPA:** Abstraction built on top of JPA/Hibernate for rapid development.