Difference between JPA, Hibernate and Spring Data JPA

|  |  |  |
| --- | --- | --- |
| Feature | JPA (Java Persistence API) | Hibernate |
|  |  | Spring Data JPA |
| Type | Specification (interface) | Implementation of JPA (also adds extra features) |
|  |  | Abstraction built on top of JPA/Hibernate |
| Defined By | Part of Java EE / Jakarta EE | Open-source (by Red Hat) |
|  |  | Part of Spring Framework |
| Provides | API for ORM (e.g., EntityManager, annotations) | Actual implementation, caching, HQL, etc. |
|  |  | Repository abstraction, query method auto-generation |
| Requires Configuration | Needs a provider (like Hibernate) to work | Yes, manually configured or via frameworks |
|  |  | Minimal — Spring Boot auto-configures most things |
| Goal | Standard way to manage relational data in Java | Complete ORM solution |
|  |  | Simplifies JPA with ready-to-use repository interfaces |
| Boilerplate Code | Yes | Less than plain JDBC but still some boilerplate |
|  |  | Much reduced (CRUD methods auto-implemented) |
| Query Language | JPQL | JPQL + HQL (Hibernate Query Language) |
|  |  | Derived queries, JPQL, Native SQL supported |
| Common Usage | In any Java app that uses ORM | Frequently used as JPA provider |
|  |  | Used with Spring Boot/Framework for easy data access |
| Example Class | @Entity, EntityManager | SessionFactory, CriteriaBuilder |
|  |  | JpaRepository, CrudRepository, @Query |