# Difference between JPA, Hibernate, and Spring Data JPA

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA |
| Type | Specification (Interface/API only) | ORM Implementation (and JPA provider) | Abstraction layer built on top of JPA + Hibernate |
| Provided By | Oracle / Jakarta EE | Red Hat | Spring Framework Team |
| Purpose | Standard for ORM in Java | Concrete ORM functionality and JPA implementation | Simplified data access using repositories |
| Requires Implementation? | Yes (Hibernate, EclipseLink, etc.) | No (it is an implementation) | No (relies on JPA providers) |
| Query Language | JPQL (Java Persistence Query Language) | HQL (Hibernate Query Language), Criteria API | Derived Query Methods, JPQL, Native SQL |
| Boilerplate Code | More (manual DAO creation) | Less (with annotations and helpers) | Least (auto-implemented methods via interface) |
| Ease of Use | Moderate | Moderate | Easiest |
| Configuration | Manual (persistence.xml) | XML or Annotation-based | Mostly application.properties (auto-configured) |
| Integration with Spring | Requires setup | Integrates via JPA adapters | Native support (Spring Boot autoconfigures everything) |
| Best Use Case | Learning ORM standards | Deep ORM control, fine-tuned configuration | Rapid development, CRUD apps with minimal setup |