# Difference between JPA, Hibernate and Spring Data JPA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature | JPA (Java Persistence API) | Hibernate | Spring Data JPA |  |  |
| Type | Specification (Interface) | Implementation (Framework) | Abstraction Layer (Built on JPA + Spring) |  |  |
| Purpose | Defines standard for ORM in Java | Provides implementation of JPA + additional features | Simplifies JPA/Hibernate usage in Spring apps |  |  |
| Package | `javax.persistence` | `org.hibernate` | `org.springframework.data.jpa.repository` |  |  |
| Usage | Provides annotations & API for ORM mapping | Implements JPA and extends with caching, lazy loading, etc. | Automatically generates repository code (e.g., CRUD) |  |  |
| Boilerplate Code | Requires manual DAO implementation | Reduces some boilerplate with APIs | Greatly reduces boilerplate using method naming conventions |  |  |
| Standalone? | No (needs implementation like Hibernate) | Yes | No (works with JPA provider like Hibernate) |  |  |
| Example | `@Entity`, `@Table`, `EntityManager` | `SessionFactory`, `CriteriaBuilder`, etc. | `CrudRepository`, `JpaRepository`, `@Query`, etc. |  |  |