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

**JPA (Java Persistence API)**

* **What it is**: A **specification (interface)** provided by Java to manage relational data in Java applications.
* **Purpose**: Defines how Java objects are mapped to database tables, and how to persist them.
* **Key Point**: JPA is just an **API**, it doesn’t provide any actual implementation.
* **Provided By**: Java (part of Java EE / Jakarta EE)

**Hibernate**

* **What it is**: A **popular implementation** of the JPA specification.
* **Purpose**: Provides the actual working code behind JPA APIs, plus extra features like caching, lazy loading, etc.
* **Key Point**: Hibernate can be used **with or without JPA**.
* **Provided By**: Red Hat

**Spring Data JPA**

* **What it is**: A **Spring-based abstraction layer** over JPA (and Hibernate).
* **Purpose**: Simplifies writing JPA data access code by **removing boilerplate code** (like DAO implementations).
* **Key Point**: You mostly write **interfaces**; Spring generates implementations at runtime.
* **Provided By**: Spring Framework

Use **JPA** for standardization.

Use **Hibernate** if you want full control and features.

Use **Spring Data JPA** if you want faster development with minimal effort.