Difference between JPA, Hibernate and Spring Data JPA

## Java Persistence API (JPA)

JPA is a specification provided by Java (JSR 338) for managing relational data using Java objects. It defines a set of interfaces and annotations to map Java objects to database tables but does not provide an implementation. Developers must use an implementation like Hibernate to use JPA in practice.

## Hibernate

Hibernate is an object-relational mapping (ORM) tool and a popular implementation of JPA. It provides functionalities such as session management, transaction handling, and query writing using HQL. However, it requires more boilerplate code such as opening sessions, beginning transactions, and handling exceptions manually.

## Example using Hibernate

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

session.save(employee);

tx.commit();

## Spring Data JPA

Spring Data JPA is a Spring-based abstraction built on top of JPA (and an implementation like Hibernate underneath). It simplifies database interactions by reducing boilerplate code, offering built-in CRUD operations, query derivation from method names, and automatic transaction management. It does not implement JPA but enhances productivity by minimizing the need to write standard repository code.

## Example using Spring Data JPA

@Autowired

private EmployeeRepository employeeRepository;

employeeRepository.save(employee);