Difference between JPA, Hibernate and Spring Data JPA

# Java Persistence API (JPA)

• JSR 338 Specification for persisting, reading and managing data from Java objects  
• Does not contain concrete implementation of the specification  
• Hibernate is one of the implementations of JPA

# Hibernate

• ORM Tool that implements JPA  
• Provides features beyond JPA (e.g., HQL, Caching)  
• Can be used independently or as a JPA provider

# Spring Data JPA

• Does not have JPA implementation, but reduces boilerplate code  
• This is another level of abstraction over JPA implementation provider like Hibernate  
• Integrated with Spring, manages transactions easily using annotations  
• Provides ready-to-use repository interfaces

# Comparison Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA |  |  |
| Type | Specification | Implementation | Abstraction over JPA |  |  |
| Requires Implementation? | Yes (like Hibernate) | No (it is an implementation) | Yes (uses Hibernate or others) |  |  |
| Boilerplate Code | Medium | High | Very Low |  |  |
| Ease of Use | Moderate | Moderate | High |  |  |
| Integrated with Spring | No | No (but can be) | Yes |  |  |
| Transaction Management | Manual/Programmatic | Manual/Programmatic | Declarative (@Transactional) |  |  |

# Code Comparison

## Hibernate Example

/\* Method to CREATE an employee in the database \*/  
public Integer addEmployee(Employee employee){  
 Session session = factory.openSession();  
 Transaction tx = null;  
 Integer employeeID = null;  
   
 try {  
 tx = session.beginTransaction();  
 employeeID = (Integer) session.save(employee);   
 tx.commit();  
 } catch (HibernateException e) {  
 if (tx != null) tx.rollback();  
 e.printStackTrace();   
 } finally {  
 session.close();   
 }  
 return employeeID;  
}

## Spring Data JPA Example

EmployeeRepository.java  
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}  
  
EmployeeService.java  
@Autowired  
private EmployeeRepository employeeRepository;  
  
@Transactional  
public void addEmployee(Employee employee) {  
 employeeRepository.save(employee);  
}

# Reference Links

https://dzone.com/articles/what-is-the-difference-between-hibernate-and-sprin-1

https://www.javaworld.com/article/3379043/what-is-jpa-introduction-to-the-java-persistence-api.html