### ✅ ****Java Persistence API (JPA)****

* **Type**: Specification (JSR 338)
* **Purpose**: Defines a standard for ORM (Object-Relational Mapping) in Java
* **Key Point**: Does **not provide an implementation**; it’s just an interface
* **Example Implementations**: Hibernate, EclipseLink
* **Common Annotations**: @Entity, @Id, @OneToMany

### ****Hibernate****

* **Type**: Implementation of JPA (also supports native features)
* **Purpose**: ORM tool that converts Java objects into database tables and vice versa
* **Key Features**:
  + Implements JPA
  + Provides additional features (e.g., caching, custom annotations)
* **Requires Manual Code**:
  + For sessions, transactions, exception handling (unless integrated with Spring)

### ✅ ****Spring Data JPA****

* **Type**: Abstraction Layer over JPA
* **Purpose**: Simplifies data access by eliminating boilerplate code
* **Depends On**: A JPA implementation (usually Hibernate)
* **Key Features**:
  + Automatic query generation (e.g. findByName)
  + Built-in support for pagination, sorting, and transactions
  + Repository interface usage (e.g. JpaRepository)

**Comparison with code examples**

#### ****Hibernate (Manual)****

public Integer addEmployee(Employee employee) {

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee); // Save object

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

**Spring Data JPA (Simplified)**

**Repository Interface:**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**Service Layer:**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee); // No session/transaction code

}

}

| Feature | JPA | Hibernate | Spring Data JPA |
| --- | --- | --- | --- |
| Type | Specification | Implementation (of JPA) | Abstraction over JPA/Hibernate |
| Boilerplate Code | Moderate | High | Low |
| Provides ORM? | No | Yes | No (uses Hibernate or other JPA providers) |
| Query Method Generation | No | No | Yes (e.g. findByName) |
| Transaction Management | Manual (or with Spring) | Manual | Automatically handled |
| Custom Repository Methods | Requires implementation | Requires implementation | Auto-generated by Spring |