**Hands on 4**

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

**🔹 JPA (Java Persistence API)**

* A **specification** (interface) for object-relational mapping (ORM).
* Does **not provide implementation**—just guidelines.
* Defines standard annotations and APIs (like @Entity, @Id, EntityManager, etc.).

### 🔹 ****Hibernate****

* A **concrete implementation of JPA**.
* Offers additional features beyond JPA, like **lazy loading, caching**, and **HQL** (Hibernate Query Language).
* Requires **manual configuration** and **boilerplate code** for session management and transactions.

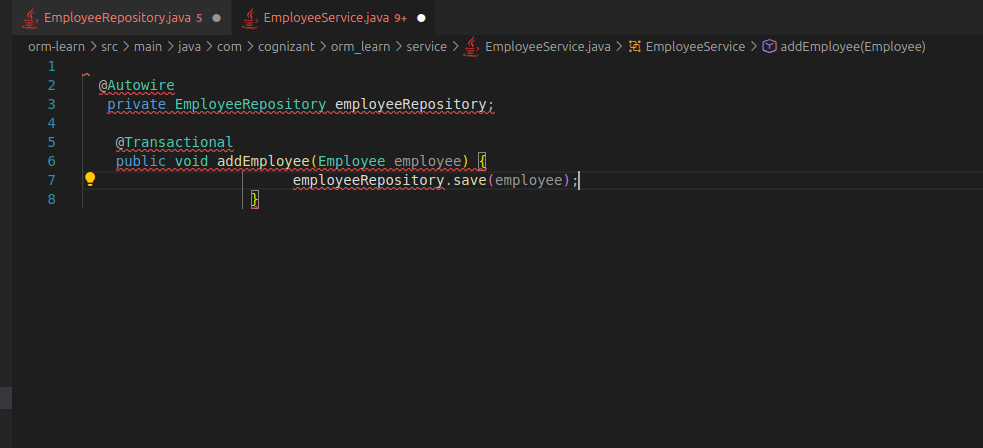
### 🔹 ****Spring Data JPA****

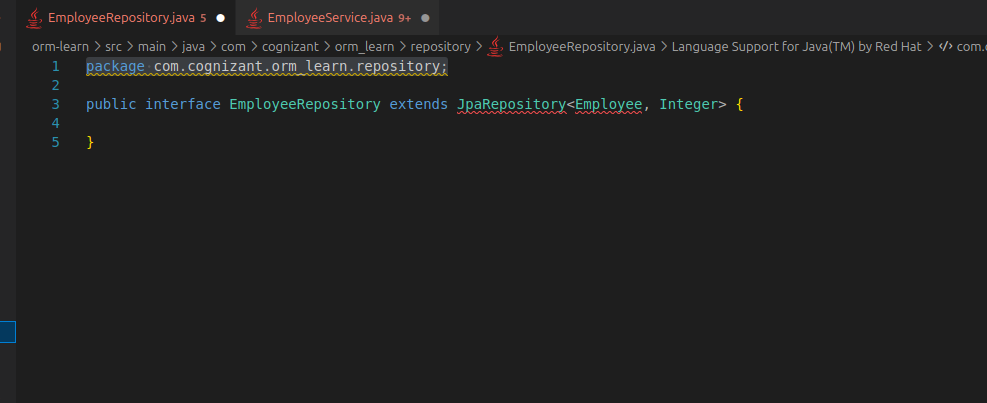
* A **Spring framework module** built on top of JPA (uses Hibernate or other providers internally).
* Reduces boilerplate using **repository interfaces** like JpaRepository.
* Automatically provides **CRUD operations**, **query generation**, and **transaction management** with minimal code.
* Simplifies integration with Spring Boot.

**Implementation :**

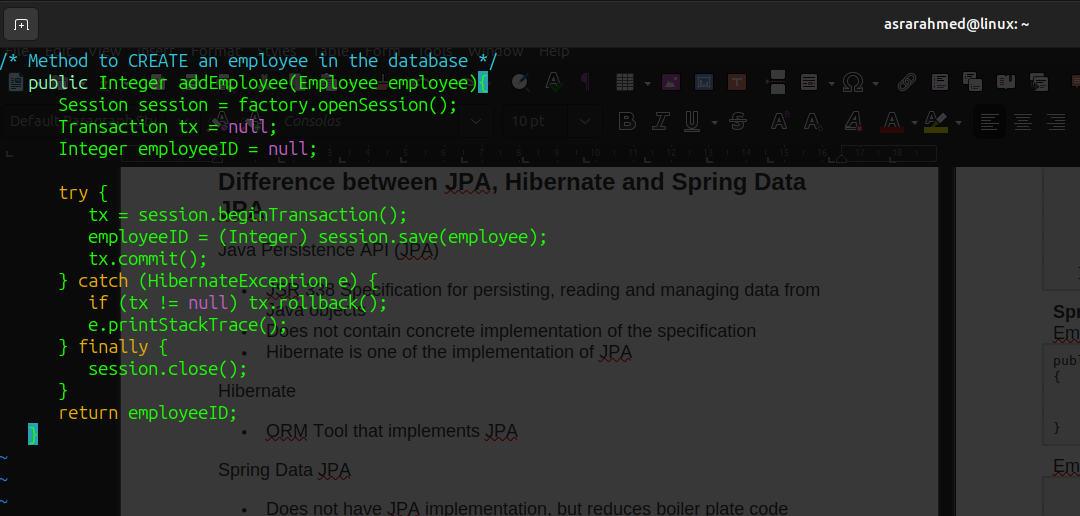
**Spring Data JPA :**

**1 EmployeeService.java**

****

****

**2 EmployeeRespository.java**

****

**Hibernate :**

### **🔸 Summary of Output Difference**

| **Aspect** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- |
| **SQL Query Logs** | **Same (uses Hibernate under the hood)** | **Same** |
| **Session/Transaction Logs** | **Visible (manual)** | **Hidden (managed by Spring)** |
| **Boilerplate Logs** | **More (open session, commit, etc.)** | **Less (handled automatically)** |
| **Output Clarity** | **Verbose** | **Clean and focused on logic** |