**Spring Boot JPA**

**Spring Boot JPA** là một phần trong hệ sinh thái Spring Data, nó tạo ra một layer ở giữa tầng service và database, giúp chúng ta thao tác với database một cách dễ dàng hơn, tự động config và giảm thiểu code thừa thãi.

**Spring Boot JPA** đã wrapper Hibernate và tạo ra một interface mạnh mẽ. Nếu như bạn gặp khó khăn khi làm việc với Hibernate thì đừng lo, bạn hãy để **Spring JPA** làm hộ.

### Cài đặt

Để thêm Spring JPA vào project, bạn cần thêm dependency spring-boot-starter-data-jpa.

Ngoài ra, để connect tới MySql, chúng ta cần driver tương ứng, vì vậy phải bổ sung thêm cả dependency mysql-connector-java vào pom.xml.

### Tạo Table và dữ liệu

### Tạo Model

@Entity

@Table(name = "user")

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

// Mapping thông tin biến với tên cột trong Database

@Column(name = "hp")

// Nếu không đánh dấu @Column thì sẽ mapping tự động theo tên biến

### Vấn đề của Hibernate truyền thống

// Giả sử đã có đối tượng session rồi

Session session = getSession();

try {

// Tất cả các lệnh hành động với DB thông qua Hibernate

// đều phải nằm trong 1 giao dịch (Transaction)

// Bắt đầu giao dịch

session.getTransaction().begin();

// Tạo một query

String sql = "Select u from " + User.class.getName() + " u ";

// Tạo đối tượng Query.

Query<User> query = session.createQuery(sql);

// Thực hiện truy vấn và lấy ra dữ liệu.

List<User> users = query.getResultList();

// In ra màn hình

for (User user : users) {

System.out.println(user);

}

// Commit dữ liệu và kết thúc session.

session.getTransaction().commit();

} catch (Exception e) {

e.printStackTrace();

// Rollback trong trường hợp có lỗi xẩy ra.

session.getTransaction().rollback();

}

### JpaRepository

Để sử dụng **Spring JPA**, bạn cần sử dụng interface JpaRepository.

Yêu cầu của interface này đó là bạn phải cung cấp 2 thông tin:

1. Entity (Đối tượng tương ứng với Table trong DB)
2. Kiểu dữ liệu của khóa chính (primary key)

@Repository

public interface UserRepository extends JpaRepository<User, Long> {

}

Vậy thôi, @Repository đánh dấu UserRepository là một Bean và chịu trách nhiệm giao tiếp với DB.

**Spring Boot** sẽ tự tìm thấy và khởi tạo ra đối tượng UserRepository trong Context. Việc tạo ra UserRepository hoàn toàn tự động và tự config, vì chúng ta đã kế thừa JpaRepository.

@Autowired

UserRepository userRepository;

userRepository.findAll()

.forEach(System.out::println);