**CÁC TẦNG MVC SPRING RESTFUL API**

# **1. Tầng Model (Domain Model / Entity Layer)**

Tầng này đại diện cho các đối tượng và dữ liệu trong ứng dụng, tức là các Entity (thực thể) sẽ ánh xạ với các bảng trong cơ sở dữ liệu.

Mỗi lớp Entity sẽ biểu diễn một bảng trong cơ sở dữ liệu và các thuộc tính của lớp tương ứng với các cột trong bảng đó.

# **2. Tầng Repository (Data Access Layer)**

Tầng này chịu trách nhiệm truy cập dữ liệu từ cơ sở dữ liệu.

Với **JPA**, bạn sẽ tạo các interface Repository để định nghĩa các thao tác cơ bản như lưu, tìm kiếm, cập nhật, và xóa.

Các Repository Interface thường kế thừa từ JpaRepository hoặc CrudRepository, giúp tự động tạo các phương thức CRUD mà không cần viết SQL thủ công.

Ngoài các phương thức có sẵn, bạn cũng có thể thêm các phương thức tùy chỉnh trong Repository bằng cách đặt tên phương thức theo mẫu mà JPA hỗ trợ. Ví dụ: findByName(String name) để tìm kiếm các bản ghi theo trường name.

Nếu cần truy vấn phức tạp, bạn có thể sử dụng **@Query** để viết truy vấn JPQL hoặc SQL trực tiếp trong Repository.

|  |
| --- |
| // Ví dụ về phương thức tùy chỉnh  List<Employee> findByDepartment(String department);  // Ví dụ truy vấn JPQL tùy chỉnh với @Query  @Query("SELECT e FROM Employee e WHERE e.salary > :salary")  List<Employee> findEmployeesWithSalaryAbove(@Param("salary") double salary); |

# **3. Tầng Service (Business Logic Layer)**

Tầng này chứa logic nghiệp vụ của ứng dụng. **Service Layer** đóng vai trò trung gian, tương tác với Repository để xử lý dữ liệu và chuẩn bị dữ liệu cho Controller.

Nếu có logic nghiệp vụ phức tạp (như các quy tắc tính toán, xác thực, hoặc xử lý đặc biệt), bạn sẽ đặt nó trong tầng Service này.

# **4. Tầng Controller (Presentation Layer)**

Tầng này nhận yêu cầu từ phía người dùng (thường qua HTTP), xử lý các yêu cầu bằng cách gọi các phương thức trong tầng Service, và trả lại dữ liệu dưới dạng JSON hoặc XML theo chuẩn RESTful.

**Controller Layer** chịu trách nhiệm về luồng dữ liệu từ và đến phía người dùng.

Đây là tầng trung gian giữa tầng **Controller** (nhận yêu cầu từ người dùng) và tầng **Repository** (tương tác với cơ sở dữ liệu). Mục tiêu của Service là giúp tổ chức mã rõ ràng và dễ bảo trì hơn bằng cách đưa logic nghiệp vụ ra khỏi tầng Controller.

# **5. Tầng Exception Handling (Exception Layer)**

Tầng này không phải là tầng bắt buộc, nhưng rất quan trọng để xử lý các lỗi có thể xảy ra trong ứng dụng và gửi thông báo lỗi một cách phù hợp cho người dùng.

Trong Spring Boot, bạn có thể tạo các lớp để xử lý ngoại lệ và trả về thông báo lỗi có cấu trúc.