

Xây dựng ứng dụng quản lý sản phẩm với Spring Boot

Mục tiêu:

- Xây dựng một ứng dụng RESTful API cơ bản sử dụng **Spring Boot**.
- Thực hiện các thao tác CRUD với cơ sở dữ liệu.
- Sử dụng mô hình **MVC** và áp dụng các khái niệm như **Dependency Injection (DI)**, **IoC**, và Spring Boot features.
- Áp dụng kiến thức về API endpoint và dữ liệu JSON.

Yêu cầu thực hiện:

1. Mô tả bài toán:

Xây dựng một ứng dụng quản lý sản phẩm bao gồm các chức năng sau:

- Thêm sản phẩm mới. (1đ)
- Cập nhật thông tin sản phẩm. (1đ)
- Xóa sản phẩm. (1đ)
- Lấy danh sách sản phẩm. (1đ)
- Tìm kiếm sản phẩm theo ID. (1đ)
- Thống kê sản phẩm theo giá. (1đ)
- Thống kê sản phẩm theo số lượng sản phẩm. (1đ)
- Tìm kiếm sản phẩm theo giá sản phẩm. (1đ)
- Tìm kiếm sản phẩm theo tên và giá sản phẩm. (1đ)

2. Yêu cầu cụ thể:

- **Entities:**
 - Tạo một **Product** entity với các thuộc tính sau:
 - **id (Integer)**: ID của sản phẩm (tự động tăng).
 - **name (String)**: Tên sản phẩm.
 - **price (Double)**: Giá sản phẩm.
 - **quantity (Integer)**: Số lượng sản phẩm.
 - **description (String)**: Mô tả sản phẩm.
- **Endpoints:**
 - **POST /api/products**: Thêm sản phẩm mới. Dữ liệu sản phẩm được gửi trong body dưới dạng JSON.
 - **PUT /api/products/{id}**: Cập nhật thông tin sản phẩm dựa trên ID.
 - **DELETE /api/products/{id}**: Xóa sản phẩm dựa trên ID.
 - **GET /api/products**: Lấy danh sách tất cả sản phẩm.
 - **GET /api/products/{id}**: Lấy thông tin một sản phẩm dựa trên ID.
- **Database:**
 - Sử dụng **H2 Database** (hoặc tự chọn một Database mà bạn tự tin, chẳng hạn như SQL Server hoặc khác).
 - Tạo bảng sản phẩm tự động bằng Spring Boot JPA (**Spring Data JPA**).
- **Validation:**

- Tên sản phẩm không được để trống.
- Giá sản phẩm phải lớn hơn 0.
- Số lượng phải là số nguyên dương.
- **Swagger:**
 - Tích hợp **Swagger** để kiểm tra các **endpoint API**.

Hướng dẫn thực hiện:

1. Cài đặt môi trường:

- Tạo một project Spring Boot mới (sử dụng **Spring Initializr** hoặc IDE như **IntelliJ/NetBeans**).
- Thêm các **dependency** cần thiết:
 - **Spring Web**.
 - **Spring Data JPA**.
 - **H2 Database** (hoặc Cơ sở dữ liệu do em lựa chọn).
 - **Spring Boot Starter Validation**.
 - **Swagger** (**Springfox** hoặc **Springdoc**).

2. Tạo cấu trúc project:

- Tạo các package theo mô hình MVC:
 - **controller**: Chứa các lớp xử lý request.
 - **service**: Chứa logic xử lý nghiệp vụ.
 - **repository**: Chứa giao tiếp với cơ sở dữ liệu.
 - **model**: Chứa các lớp entity.

3. Triển khai chức năng:

- Tạo lớp Product trong package model với các annotation như **@Entity**, **@Id**, **@GeneratedValue**.
- Tạo interface **ProductRepository** trong package **repository** và kế thừa **JpaRepository**.
- Tạo lớp **ProductService** trong package **service** để xử lý các logic liên quan đến sản phẩm.
- Tạo lớp **ProductController** trong package **controller** để định nghĩa các API endpoint.

4. Tích hợp Swagger:

- Cấu hình Swagger trong file **application.properties** hoặc tạo một lớp cấu hình riêng.

5. Kiểm tra ứng dụng:

- Chạy ứng dụng **Spring Boot**.
- Sử dụng **Swagger UI** hoặc công cụ như **Postman** để kiểm tra các API.

Nộp bài:

- Nén toàn bộ project thành file .zip (tên file là tên của em viết liền không dấu) và gửi lại (kể cả tệp tin PDF và video).
- Nội dung nộp bao gồm:

- Mã nguồn đầy đủ (**code**).
- Tài liệu mô tả cách chạy ứng dụng (**Nộp bằng file PDF**).
- Video minh họa kết quả thực hiện các yêu cầu trên từ 1 đến 9 (**Nộp video**).

Lưu ý: (1đ) trình bày nộp đầy đủ theo yêu cầu. Nếu bị thiếu file hoặc không đúng định dạng theo yêu cầu cần nộp thì bị trừ 2 điểm.