Tìm hiểu về java spring boot

Tìm hiểu về java core

- **Configuration**:
    - Khái niệm: Spring Core Configuration là quá trình định cấu hình Spring Framework, bao gồm việc chỉ định các chi tiết cấu hình khác nhau cần thiết để ứng dụng hoạt động bình thường. Điều này có thể bao gồm thiết lập các bean, chỉ định các thành phần phụ thuộc của bean, định cấu hình các khía cạnh lập trình hướng theo khía cạnh (AOP), v.v. Cấu hình có thể được thực hiện thông qua mã Java, tệp XML hoặc sử dụng chú thích trong mã.
    - Các tệp cấu hình có đuôi .prooerties thường nằm trong thư mục src/main/resources
    - Import các file dữ liệu sử dung @import vào MainApplycation.java
    - Cấu hình các file config với xml
- **Dependence Injection:**
    - Khái niệm: DI (Dependency Injection) trong Java Spring là một cơ chế cho phép ta inject (tiêm) các dependency (các đối tượng phụ thuộc) vào trong một đối tượng khác, mà không cần phải tạo chúng bằng tay.
    - Thực hiện DI bao gồm Constructer, Setter, Field injection
        - Cụ thể, Constructor Injection là kỹ thuật chuyển giao phụ thuộc bằng cách truyền chúng vào qua constructor của đối tượng.
        - Setter Injection là kỹ thuật chuyển giao phụ thuộc bằng cách sử dụng các phương thức setter để thiết lập các giá trị của các thuộc tính.
        - Field Injection là kỹ thuật chuyển giao phụ thuộc bằng cách sử dụng các thuộc tính trực tiếp.
- **IOC spring (Inversion of control)**
    - Khái niệm: Spring Container (IoC Container) sẽ tạo các đối tượng, lắp rắp chúng lại với nhau, cấu hình các đối tượng và quản lý vòng đời của chúng từ lúc tạo ra cho đến lúc bị hủy.
    - Một số tính chất của ioc:
        - Đảo ngược quyền kiểm soát (Inversion of Control): đối tượng sẽ được container tạo và quản lý, không phải được tạo và quản lý bởi tầng ứng dụng.
        - Phụ thuộc vào interface (Dependency on Interfaces): các đối tượng được container tạo ra đều phụ thuộc vào các interface, không phải phụ thuộc vào các đối tượng cụ thể.
        - Tái sử dụng (Reuse): việc quản lý đối tượng bởi container giúp tái sử dụng các đối tượng một cách dễ dàng và hiệu quả.
        - Kết nối (Wiring): các phụ thuộc giữa các đối tượng được quản lý bởi container.
        - Định nghĩa (Definition): các đối tượng được định nghĩa trong file cấu hình, dễ dàng thay đổi và tùy biến.
- **AOP**
    - Khái niệm: AOP (Aspect Oriented Programming) là một phương pháp lập trình hướng đối tượng cho phép chúng ta tách các vấn đề quan trọng khác nhau và tập trung vào một chức năng cụ thể của hệ thống. AOP cho phép chúng ta định nghĩa các khía cạnh (aspects) để tách các khía cạnh đó ra khỏi mã chương trình chính.
    - Các khái niệm vê thuật ngữ AOP:
        - Aspect: Aspect(khía cạnh) là một module đóng gói những advice và các pointcut và cung cấp tính xuyên suốt. Một ứng dụng có thể có bất

kỳ khía cạnh nào. Chúng ta có thể triển khai một khía cạnh bằng cách sử dụng class thông thường với annotation @Aspect.
        - Advice: Advice là một hành động mà chúng ta thực hiện trước hoặc sau khi method thực hiện. Có 5 loại advice trong Spring AOP framework đó là: before, after, after-returning, after-throwing, and around advice.
        - Pointcut: pointcut là một biểu thức chọn một hoặc nhiều điểm nối(join poin) nơi advice được thực thi. Chúng ta có thể xác định các pointcut bằng cách sử dụng các biểu thức hoặc các mẫu. Nó sử dụng các loại biểu thức khác nhau phù hợp với các điểm tham gia.
        - Target object: Đối tượng mà được các advice sử dụng thì được gọi là target object. Target object thì luôn là một proxy. Nó có nghĩa là một lớp con được tạo ra tại thời điểm chạy, trong đó phương thức đích bị ghi đè và các advice được đưa vào dựa trên cấu hình của chúng.
        - Join point: Join point là một điểm trong ứng dụng nơi chúng ta áp dụng khía cạnh AOP.
- **Spring MVC:**
    - Model chứa dữ liệu của ứng dụng. Dữ liệu có thể là một đối tượng đơn lẻ hoặc một tập hợp các đối tượng.
    - Controller chứa logic nghiệp vụ của một ứng dụng. Ở đây, chú thích @Controller được sử dụng để đánh dấu lớp là bộ điều khiển.
    - View biểu thị thông tin được cung cấp ở một định dạng cụ thể. Nói chung, JSP+JSTL được sử dụng để tạo một trang xem.
- **Annotation:**
    - Khái niệm: là một dạng siêu dữ liệu cung cấp dữ liệu về một chương trình. Nói cách khác, chú thích được sử dụng để cung cấp thông tin bổ sung về một chương trình.
    - Một vài anotation phổ biến:
        - @SpringBootApplication tự động scan các cấu hình spring và tự động quét các thành phần spring.
        - @RestController cung cấp việc xây dựng api dễ dàng hơn.
        - @Autowired là một annotation trong Java Spring framework được sử dụng để thực hiện Dependency Injection. Nó được sử dụng để tiêm các đối tượng đã được khởi tạo từ container (hay còn gọi là Spring Context) vào các bean (component, service, controller...) khác để đảm bảo tính kết nối giữa các bean trong ứng dụng.
        - @Bean cho biết rằng một method tạo ra một bean sẽ được quản lý bởi Spring container. Nó là một annotation ở method. Trong khi cấu hình Java (@Configuration), phương thức được thực thi và giá trị trả về của nó được đăng ký dưới dạng bean trong BeanFactory.
- **Các lớp trong spring boot**
    - Presentation Layer: xử lý các yêu cầu HTTP, dịch tham số JSON thành đối tượng và xác thực yêu cầu rồi chuyển nó sang lớp nghiệp vụ.
    - Business Layer: Tầng nghiệp vụ xử lý tất cả logic nghiệp vụ. Nó bao gồm các lớp dịch vụ và sử dụng các dịch vụ được cung cấp bởi các lớp truy cập dữ liệu. Nó cũng thực hiện ủy quyền và xác nhận.
    - Persistence Layer: chứa tất cả logic lưu trữ và dịch các đối tượng nghiệp vụ từ và sang các hàng cơ sở dữ liệu.
    - Database Layer thực hiện thao tác với cơ sở dữ liệu