**Logging trong Spring Boot**

* Khi làm việc với ghi log trong Java, có nhiều thư viện khác nhau mà chúng ta có thể sử dụng:
  + java.util.logging: Là thư viện ghi log mặc định đi kèm với Java. Nó cung cấp một API đơn giản để ghi log và có sẵn trong JDK. Tuy nhiên, nó có hạn chế về tính linh hoạt và khả năng tùy chỉnh so với các thư viện khác
  + Log4j2: Là phiên bản tiếp theo của Log4j và là một thư viện ghi log mạnh mẽ và linh hoạt. Nó cung cấp khả năng tùy chỉnh cao và hỗ trợ nhiều định dạng ghi log. Log4j2 cũng có khả năng xử lý các thông báo log ở cấp độ thấp hơn (DEBUG, TRACE) so với các thư viện khác
  + SLF4J (Simple Logging Facade for Java): Là một facade (giao diện) đơn giản để làm việc với các thư viện ghi log khác như Log4j, java.util.logging, Logback và nhiều thư viện khác. SLF4J cho phép chúng ta viết mã ghi log không phụ thuộc vào một thư viện cụ thể, giúp giảm sự ràng buộc và dễ dàng chuyển đổi giữa các thư viện ghi log khác nhau
  + Logback: Là một thư viện ghi log mạnh mẽ, linh hoạt và hiệu suất cao. Đây cũng là thư viện được khuyến nghị khi sử dụng SLF4J. Logback hỗ trợ nhiều tính năng như lọc log, tạo bản ghi tự động, quản lý file log và nhiều tính năng khác
* Nếu chúng ta sử dụng SLF4J, chúng ta cần thêm dependency của một implementation cụ thể của SLF4J. SLF4J chỉ là một facade và không cung cấp chức năng ghi log thực tế, mà chỉ cung cấp một API chung để tương tác với các thư viện ghi log khác
  + Với Logback:

A computer screen shot of a program code

Description automatically generated

* + Với Log4j2:

A screenshot of a computer program

Description automatically generated

* Mặc định trong dự án Spring Boot, SLF4J, Logback và Log4j đều được tích hợp trong *spring-boot-starter-web*, và Logback được sử dụng làm implementation mặc định của SLF4J
* **Phạm vi quét logging trong Spring Boot**
  + Root: áp dụng cho tất cả các logger
  + Package: áp dụng cho các logger trong package
  + Class: áp dụng cho các logger trong class

**A close-up of a number

Description automatically generated**

* **Spring Boot Logging Level**
  + Spring Boot Logging Level là cách để cấu hình mức độ ghi log cho ứng dụng Spring Boot
  + Các mức độ / level ghi log từ thấp đến cao:

|  |  |
| --- | --- |
| Mức độ | Nội dung |
| TRACE | Mức độ thấp nhất, ghi lại tất cả các hoạt động của ứng dụng, bao gồm cả thông tin rất chi tiết và nhỏ nhặt |
| DEBUG | Ghi lại các thông tin debug, chẳng hạn như các giá trị biến, lỗi runtime,... |
| INFO | Mức độ trung bình, ghi lại các thông tin quan trọng về hoạt động của ứng dụng, chẳng hạn như thông tin về các hoạt động chính của ứng dụng |
| WARN | Ghi lại các thông báo cảnh báo về những tình huống không mong muốn xảy ra, nhưng vẫn có thể xử lý được |
| ERROR | Mức độ cao nhất, ghi lại các lỗi xảy ra trong quá trình chạy ứng dụng, như lỗi kết nối cơ sở dữ liệu, lỗi trong code, lỗi hệ thống,... |

* + Khi chúng ta không cấu hình giá trị cho thuộc tính logging.level trong Spring Boot, thư viện logging mà chúng ta đang sử dụng (Logback, Log4j2, …) sẽ quyết định cấp độ log mặc định với phạm vi là root. Ví dụ, nếu chúng đang sử dụng Logback, thư viện này có thể sử dụng INFO làm cấp độ log mặc định. Điều này có nghĩa là các thông báo log ở cấp độ INFO và cấp độ cao hơn sẽ được xuất ra (ví dụ như WARN, ERROR), trong khi các cấp độ log thấp hơn (TRACE, DEBUG) sẽ không được xuất ra
* **Demo trong Spring Boot**

A close-up of a website

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer program

Description automatically generated

Lưu ý:

A screenshot of a computer

Description automatically generated

* **Spring Boot Log Format**
  + Định dạng log của Spring Boot là cách thông tin được ghi lại trong quá trình logging trong ứng dụng. Nó bao gồm các trường thông tin như cấp độ log, thời gian, tên class, nội dung log và các thông tin khác có thể tùy chỉnh
  + Spring Boot không cung cấp trực tiếp các định dạng log format riêng cho việc cấu hình logging. Các định dạng log format chính xác sẽ phụ thuộc vào thư viện logging mà chúng ta sử dụng (mặc định là Logback)
  + Nếu chúng ta muốn tùy chỉnh định dạng log format trong ứng dụng Spring Boot, chúng ta cần tạo tệp cấu hình cho thư viện logging chúng ta đang sử dụng (ví dụ: "logback-spring.xml" cho Logback, "log4j2.xml" cho Log4j2) và sử dụng các encoder, layout hoặc định dạng log format cụ thể mà thư viện đó hỗ trợ
  + Ví dụ định dạng log dưới dạng JSON với logback:

Cấu hình file logback-spring.xml:

A computer code with text

Description automatically generated

Encoder:

A computer code with black text

Description automatically generated

Demo:

A computer code with text

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Spring Boot Logging File**
  + Khi chúng ta không thực hiện bất kỳ cấu hình logging nào, Spring Boot sử dụng một cấu hình logging mặc định và thông báo log sẽ được hiển thị trên console
  + Spring Boot Logging File là việc ghi các thông tin log vào các tệp tin (file log) thay vì xuất log lên console hoặc các nguồn khác. Việc ghi log vào file là một cách tiện lợi và hữu ích để theo dõi và phân tích hoạt động của ứng dụng trong môi trường sản phẩm hoặc trên các máy chủ không có giao diện người dùng. Bằng cách lưu trữ log vào file, chúng ta có thể kiểm tra và xem lại các thông tin log sau này để giải quyết các vấn đề xảy ra trong hệ thống
  + Demo:

A screenshot of a computer

Description automatically generated

A computer code with text

Description automatically generated

A close-up of a computer code

Description automatically generated

* **Sử dụng @Slf4j của Lombok**

A screen shot of a computer code

Description automatically generated