Lazy Initialization

Initialization – Khởi tạo

- Khi Spring Application chạy, tất cả các Bean sẽ được khởi tạo
- Tất cả các @Component hoặc annotation khác sẽ được khởi tạo và sẵn sàng cung cấp cho các vị trí cần thiết

Thử nghiệm lại ví dụ có dùng Qualifier

```
no usages
@Component
public class EmailService implements MessageInterface{

    no usages
    public EmailService(){
        System.out.println("Constructor of " +getClass().getSimpleName());
    }

    1 usage
    @Override
    public String sendMessage() { return "Email sending ..... 123456 "; }
}
```

Lazy Initialization

- Lazy initialization là một kỹ thuật được sử dụng trong Spring Framework để tạo ra các đối tượng bean chỉ khi cần thiết, thay vì tạo tất cả các đối tượng bean ngay từ đầu khi ứng dụng khởi động. Kỹ thuật này mang lại một số lợi ích quan trọng trong việc quản lý tài nguyên và tối ưu hóa hiệu suất ứng dụng. Dưới đây là một số lợi ích của Lazy Initialization trong Spring:

@Lazy

- Khởi tạo Lazy ➔ Khởi tạo Bean khi cần thiết
- Bean được tạo ra khi:
  - Nó cần để tiêm vào phụ thuộc
  - Một yêu cầu cụ thể

Cách cấu hình Lazy

- Cách 1:

  Đối với từng class thêm annotation @Lazy

- Cách 2:

  Cấu hình cho toàn bộ dự án tại file application.properties:

  spring.main.lazy-initialization=true

```java
@Service
@Lazy
public class MyService {
    public MyService() {
        System.out.println("MyService initialized");
    }

    // Các phương thức khác của MyService
}
```

```
spring.main.lazy-initialization=true
```

Lợi ích

- Tiết kiệm tài nguyên: Khi sử dụng lazy initialization, các đối tượng bean chỉ được tạo khi thực sự cần thiết, giúp tiết kiệm tài nguyên hệ thống. Điều này đặc biệt hữu ích khi ứng dụng của bạn có hàng trăm hoặc hàng ngàn đối tượng bean và không phải tất cả chúng đều được sử dụng trong mọi tình huống.
- Tăng tốc độ khởi động ứng dụng: Khi sử dụng lazy initialization, Spring sẽ chỉ tạo các đối tượng bean khi cần thiết, giúp giảm thời gian khởi động ứng dụng. Điều này đặc biệt quan trọng đối với các ứng dụng lớn và phức tạp, nơi việc tạo tất cả các đối tượng bean ngay từ đầu có thể mất nhiều thời gian.
- Hỗ trợ việc quản lý trạng thái: Lazy initialization giúp hạn chế việc lưu trữ và quản lý trạng thái của các đối tượng bean trong suốt quá trình chạy ứng dụng. Thay vì giữ tất cả các đối tượng bean trong bộ nhớ ngay từ đầu, chỉ có những đối tượng bean cần thiết được tạo và duy trì trạng thái của chúng. Điều này giúp giảm rủi ro lỗi và tăng tính ổn định của ứng dụng.
- Hỗ trợ tải chậm dữ liệu: Khi sử dụng lazy initialization, bạn có thể tải dữ liệu từ các nguồn bên ngoài, chẳng hạn như cơ sở dữ liệu hoặc dịch vụ web, chỉ khi cần thiết. Điều này hữu ích khi dữ liệu có kích thước lớn và việc tải tất cả dữ liệu ngay từ đầu sẽ tốn thời gian và tài nguyên.
- Tích hợp tốt với các quyền kiểm soát truy cập: Lazy initialization cho phép bạn kiểm soát quyền truy cập đến các đối tượng bean. Bạn có thể xác định rằng chỉ có một số phần của ứng dụng mới được truy cập và tạo các đối tượng bean liên quan chỉ khi cần thiết.

Lưu ý

- Khi sử dụng trong @RestController, Bean sẽ không được tạo ra cho đến khi Request được gọi
- Cần đảm bảo đủ Memory khi khởi tạo đối tượng
- Một số lỗi sẽ không được phát hiện kịp thời

Thực hành

- Xây dựng lại ví dụ gửi tin nhắn
- Cấu hình Lazy Initialization bằng cả 2 cách