**Exercise 11: Online Bookstore - Integrating Spring Boot Actuator**

1. **Add Actuator Dependency**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

**2. Expose and Customize Actuator Endpoints**

Actuator exposes various endpoints that allow you to monitor and interact with your application, such as /health, /metrics, /info, etc. You can enable and customize these endpoints through your application.properties or application.yml.

**Enable Actuator Endpoints in application.properties:**

# Enable all Actuator endpoints

management.endpoints.web.exposure.include=\*

# Restrict sensitive endpoints (if needed)

management.endpoints.web.exposure.exclude=shutdown

Enable Actuator Endpoints in application.yml (alternative):

management:

endpoints:

web:

exposure:

include: "\*"

exclude: "shutdown"

1. **Expose Custom Metrics**

import io.micrometer.core.instrument.MeterRegistry;

import org.springframework.stereotype.Component;

@Component

public class CustomMetrics {

private final BookService bookService;

public CustomMetrics(MeterRegistry registry, BookService bookService) {

this.bookService = bookService;

// Register a custom gauge metric for counting books

registry.gauge("bookstore.book.count", this, CustomMetrics::getBookCount);

}

public int getBookCount() {

return bookService.getAllBooks().size();

}

}

1. **Security Considerations**

management.endpoint.shutdown.enabled=true

management.security.enabled=true

5. **Testing Actuator Endpoints**