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

**Business Scenario:**

Monitor and manage your bookstore's RESTful services using Spring Boot Actuator.

**Instructions:**

1. **Add Actuator Dependency:**
   * Include the Spring Boot Actuator dependency in your project.
2. **Expose Actuator Endpoints:**
   * Enable and customize Actuator endpoints.
3. **Custom Metrics:**

Expose custom metrics for monitoring your application

**pom.xml**

<dependency>

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

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

</dependency>

**application.properties**

# Enable all Actuator endpoints

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

# Customize the base path for Actuator endpoints

management.endpoints.web.base-path=/actuator

# Customize the server port for Actuator

management.server.port=8081

# Enable specific endpoints (e.g., health, info)

management.endpoint.health.enabled=true

management.endpoint.info.enabled=true

# Example: Security configuration (if necessary)

# management.endpoint.shutdown.enabled=true

**CustomMetricsService.java**

package com.example.bookstoreapi.metrics;

import io.micrometer.core.instrument.MeterRegistry;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import javax.annotation.PostConstruct;

@Service

public class CustomMetricsService {

private final MeterRegistry meterRegistry;

@Autowired

public CustomMetricsService(MeterRegistry meterRegistry) {

this.meterRegistry = meterRegistry;

}

@PostConstruct

public void initMetrics() {

// Custom metric example: counting the number of books added

meterRegistry.counter("bookstore.books.added.count").increment(0);

// Another example: gauge to monitor the number of active users

meterRegistry.gauge("bookstore.users.active.count", 0);

}

public void incrementBookCount() {

meterRegistry.counter("bookstore.books.added.count").increment();

}

public void updateActiveUserCount(int count) {

meterRegistry.gauge("bookstore.users.active.count", count);

}

}

**BookService.java**

package com.example.bookstoreapi.service;

import com.example.bookstoreapi.metrics.CustomMetricsService;

import com.example.bookstoreapi.model.Book;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class BookService {

private List<Book> books = new ArrayList<>();

@Autowired

private CustomMetricsService customMetricsService;

public Book getBookById(Long id) {

return books.stream().filter(book -> book.getId().equals(id)).findFirst().orElse(null);

}

public List<Book> getAllBooks() {

return books;

}

public Book createBook(Book book) {

books.add(book);

customMetricsService.incrementBookCount(); // Increment the custom book count metric

return book;

}

public Book updateBook(Long id, Book book) {

Book existingBook = getBookById(id);

if (existingBook != null) {

existingBook.setTitle(book.getTitle());

existingBook.setAuthor(book.getAuthor());

}

return existingBook;

}

public void deleteBook(Long id) {

books.removeIf(book -> book.getId().equals(id));

}

}