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

**build.gradle (or pom.xml)**

dependencies {

implementation 'org.springframework.boot:spring-boot-starter-actuator'

}

**application.properties (or application.yml)**

properties

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

**BookstoreMetrics**

package com.bookstore.api.metrics;

import io.micrometer.core.instrument.Counter;

import io.micrometer.core.instrument.MeterRegistry;

import org.springframework.stereotype.Component;

@Component

public class BookstoreMetrics {

private final Counter booksCreated;

private final Counter booksUpdated;

private final Counter booksDeleted;

public BookstoreMetrics(MeterRegistry meterRegistry) {

booksCreated = meterRegistry.counter("books.created");

booksUpdated = meterRegistry.counter("books.updated");

booksDeleted = meterRegistry.counter("books.deleted");

}

public void incrementBooksCreated() {

booksCreated.increment();

}

public void incrementBooksUpdated() {

booksUpdated.increment();

}

public void incrementBooksDeleted() {

booksDeleted.increment();

}

}

**BookController**

package com.bookstore.api.controllers;

import com.bookstore.api.dtos.BookDTO;

import com.bookstore.api.entities.Book;

import com.bookstore.api.mappers.BookMapper;

import com.bookstore.api.metrics.BookstoreMetrics;

import com.bookstore.api.services.BookService;

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

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/books")

public class BookController {

private final BookService bookService;

private final BookMapper bookMapper;

private final BookstoreMetrics bookstoreMetrics;

@Autowired

public BookController(BookService bookService, BookMapper bookMapper, BookstoreMetrics bookstoreMetrics) {

this.bookService = bookService;

this.bookMapper = bookMapper;

this.bookstoreMetrics = bookstoreMetrics;

}

@PostMapping

public ResponseEntity<BookDTO> createBook(@RequestBody @Valid BookDTO bookDTO) {

Book book = bookMapper.bookDTOToBook(bookDTO);

book = bookService.createBook(book);

bookstoreMetrics.incrementBooksCreated();

return new ResponseEntity<>(bookMapper.bookToBookDTO(book), HttpStatus.CREATED);

}

@PutMapping("/{id}")

public ResponseEntity<BookDTO> updateBook(@PathVariable Long id, @RequestBody @Valid BookDTO bookDTO) {

Book book = bookMapper.bookDTOToBook(bookDTO);

book = bookService.updateBook(id, book);

bookstoreMetrics.incrementBooksUpdated();

return new ResponseEntity<>(bookMapper.bookToBookDTO(book), HttpStatus.OK);

}

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteBook(@PathVariable Long id) {

bookService.deleteBook(id);

bookstoreMetrics.incrementBooksDeleted();

return new ResponseEntity<>(HttpStatus.NO\_CONTENT);

}

}

In the above code, i have integrated Spring Boot Actuator into the bookstore application.

- I have added the Spring Boot Actuator dependency to the project.

- I have enabled and customized Actuator endpoints.

- I have exposed custom metrics for monitoring the application.

Note that i have created a custom metrics class BookstoreMetrics to expose metrics for book creation, update, and deletion. We have also injected this class into the BookController to increment the metrics accordingly.