<dependency>

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

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

</dependency>

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

# Expose all Actuator endpoints

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

# Customize the base path for Actuator endpoints

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

# Enable detailed health information

management.endpoint.health.show-details=always

curl <http://localhost:8080/actuator/health>

curl <http://localhost:8080/actuator/info>

curl <http://localhost:8080/actuator/metrics>

import io.micrometer.core.instrument.MeterRegistry;

import org.springframework.stereotype.Component;

import javax.annotation.PostConstruct;

@Component

public class CustomMetrics {

private final MeterRegistry meterRegistry;

public CustomMetrics(MeterRegistry meterRegistry) {

this.meterRegistry = meterRegistry;

}

@PostConstruct

public void init() {

// Custom counter metric

meterRegistry.counter("books.sold", "type", "fiction").increment();

// Custom gauge metric

meterRegistry.gauge("books.available", this, CustomMetrics::getAvailableBooks);

}

public int getAvailableBooks() {

// Logic to return the number of available books

return 100; // Example value

}

}

curl <http://localhost:8080/actuator/metrics/books.sold>