<dependency>

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

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

</dependency>

# Enable specific actuator endpoints

management.endpoints.web.exposure.include=health,info,metrics

# Customize the base path for actuator endpoints

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

# Enable detailed metrics (optional)

management.endpoint.metrics.enabled=true

management:

endpoints:

web:

exposure:

include: health,info,metrics

base-path: /actuator

endpoint:

metrics:

enabled: true

import io.micrometer.core.instrument.MeterRegistry;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class CustomMetricsConfig {

@Bean

public CustomMetrics customMetrics(MeterRegistry meterRegistry) {

return new CustomMetrics(meterRegistry);

}

public static class CustomMetrics {

public CustomMetrics(MeterRegistry meterRegistry) {

// Register a custom counter

meterRegistry.counter("custom.bookstore.orders", "status", "completed");

// Register a custom gauge

meterRegistry.gauge("custom.bookstore.inventory.size", new InventorySizeGauge());

}

private static class InventorySizeGauge implements Gauge {

@Override

public Number value() {

// Return the current size of the bookstore's inventory

return getInventorySize();

}

private int getInventorySize() {

// Logic to retrieve inventory size

return 100; // Example value

}

}

}

}