Spring Boot Actuator Lab Documentation

This document covers the complete implementation of the Spring Boot Actuator Lab. It includes enabling Actuator, exploring default endpoints, creating custom endpoints, securing endpoints, and optional integration with external monitoring tools like Prometheus.

Project Overview:

- This Spring Boot application is a CRUD system for managing books, integrated with Spring Actuator to monitor the application’s performance, health, and metrics.
- The Actuator was enabled to expose default endpoints, and a custom endpoint was created to track the number of books stored in the database.
- Sensitive Actuator endpoints were secured, and an optional integration with Prometheus was demonstrated.

Here is the complete documentation you can copy directly for your use:

Spring Boot Actuator Lab Documentation

This document covers the complete implementation of the Spring Boot Actuator Lab. It includes enabling Actuator, exploring default endpoints, creating custom endpoints, securing endpoints, and optional integration with external monitoring tools like Prometheus.

Lab Objectives:

1. Understand the role of Spring Actuator in monitoring Spring Boot applications.
2. Utilize Actuator endpoints to gather application metrics and health information.
3. Explore advanced Actuator features and customization.
4. Integrate with external monitoring tools (optional).

Project Overview:

- This Spring Boot application is a CRUD system for managing books, integrated with Spring Actuator to monitor the application’s performance, health, and metrics.
- The Actuator was enabled to expose default endpoints, and a custom endpoint was created to track the number of books stored in the database.
- Sensitive Actuator endpoints were secured, and an optional integration with Prometheus was demonstrated.

Implementation Summary:

1. **Enabling Spring Actuator:**
   - Added `spring-boot-starter-actuator` to the `pom.xml`.
   - Configured Actuator endpoints in `application.properties`

Java

```
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
```

- This allows full access to default Actuator endpoints, including health, metrics, and environment information.

Default Endpoints:

- `/actuator/health`: Returns the health status of the application.
- `/actuator/metrics`: Provides various metrics like memory usage, CPU load, and more.
- `/actuator/info`: Displays application-specific information.
- Demonstration was done using Postman to retrieve the responses from these endpoints.

Custom Endpoint:

- A custom endpoint `/actuator/bookCount` was created to show the number of books in the database.

External Monitoring Integration (Optional):

- Actuator metrics were integrated with Prometheus for monitoring.
- Configured Prometheus to scrape metrics from the `/actuator/prometheus` endpoint.
- The scraped metrics can be visualized using Grafana for real-time application monitoring.