Spring Boot Actuator

Spring Boot Actuator is a powerful tool that provides built-in production-ready features to help monitor and manage applications. These features allow developers to view application health, metrics, environment
settings, and more, making it easier to maintain and troubleshoot production environments.

Health Endpoints

Health checks are used to determine if an application is running properly. The /actuator/health endpoint
provides information about the application's health. Custom Health indicators can be implemented using
HealthIndicator and added to the health status.

Metrics

Metrics help monitor various aspects of the application such as memory usage, CPU usage, and more.
Default metrics like JVM memory usage and uptime are provided through /actuator/metrics, while custom
metrics can be added using the MeterRegistry bean.

Environment and Config Properties

The /actuator/env endpoint exposes current environment variables and application configuration properties.
This is useful for debugging configuration issues and allows viewing of active profiles and properties.

Thread Dumps

The /actuator/threaddump endpoint captures thread information for troubleshooting potential performance
bottlenecks or deadlocks. It is particularly helpful for diagnosing multi-threaded issues.

Loggers

The /actuator/loggers endpoint allows you to view and configure logging levels for various packages at
runtime. Logging levels can be adjusted dynamically without redeploying the application.

Auditing

Spring Boot Actuator integrates auditing features to track security events (e.g., user logins or access control
violations). Custom auditing events can be added by implementing AuditEventRepository.

Info Endpoint

The /actuator/info endpoint displays arbitrary application information such as version or build details. Custom information can be added using the info section in application.properties.

Customizing Actuator

Custom Endpoints

I can create custom Actuator endpoints by implementing @Endpoint for specific needs, such as providing application-specific diagnostics.

Best Practices for Using Spring Actuator

Secure Your Endpoints

Always secure sensitive Actuator endpoints with Spring Security

Limit Exposure in Production

Only expose necessary endpoints in production environments. Avoid exposing sensitive endpoints like /env and /loggers unless absolutely required.

Use of Custom Health Indicators

Add application-specific health indicators to ensure detailed health reporting for business-critical components like databases, external APIs, or message brokers.

Regular Audits

Regularly audit Actuator logs and metrics to proactively identify and resolve potential issues before they impact users.