

# Spring Actuator: Features and Best Practices

## Introduction

Spring Boot Actuator is a sub-project of Spring Boot that adds several production-ready features to your application. It provides built-in endpoints and tools for monitoring and managing your Spring Boot application.

## Setup

To add Spring Actuator to your project, include the following dependency in your `pom.xml`:

xml

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>
```

Configure Actuator in your `application.properties` or `application.yml`:

### properties

Enable all Actuator endpoints

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

Customize base path for actuator endpoints (optional)

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

## Key Features and Code Examples

### 1. Health Checks

- Provides information about application health
- Customizable for specific needs

#### Example of a custom health indicator:

```
```java
import org.springframework.boot.actuate.health.Health;
import org.springframework.boot.actuate.health.HealthIndicator;
import org.springframework.stereotype.Component;

@Component
public class CustomHealthIndicator implements HealthIndicator {

    @Override
    public Health health() {
        int errorCode = check(); // perform some specific health check
        if (errorCode != 0) {
            return Health.down().withDetail("Error Code", errorCode).build();
        }
        return Health.up().build();
    }

    private int check() {
        // Your logic to check health
        return 0;
    }
}
```
```

## 2. Metrics

- Exposes various metrics about the application
- Integrates with monitoring systems like Prometheus

Example of creating a custom metric:

```
```java
import io.micrometer.core.instrument.Counter;
import io.micrometer.core.instrument.MeterRegistry;
import org.springframework.stereotype.Service;

@Service
public class CustomMetricService {
    private final Counter customCounter;

    public CustomMetricService(MeterRegistry meterRegistry) {
        this.customCounter = Counter.builder("custom.metric")
            .description("A custom metric")
            .register(meterRegistry);
    }

    public void incrementCustomMetric() {
        this.customCounter.increment();
    }
}
```
```

## Info Endpoint

- Displays arbitrary application information

### Example of customizing the info endpoint:

```
import org.springframework.boot.actuate.info.Info;
import org.springframework.boot.actuate.info.InfoContributor;
import org.springframework.stereotype.Component;

@Component
public class CustomInfoContributor implements InfoContributor {

    @Override
    public void contribute(Info.Builder builder) {
        builder.withDetail("example", "This is custom info")
            .withDetail("version", "1.0.0");
    }
}
```

## Best Practices

### 1. Security

- Secure actuator endpoints, especially in production
- Use Spring Security to control access

### Example of securing Actuator endpoints:

**java**

```
import org.springframework.boot.actuate.autoconfigure.security.servlet.EndpointRequest;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapt
er;
```

@Configuration

```
public class ActuatorSecurityConfig extends WebSecurityConfigurerAdapter {
```

@Override

```
protected void configure(HttpSecurity http) throws Exception {
```

```
    http.requestMatcher(EndpointRequest.toAnyEndpoint())
```

```
        .authorizeRequests()
```

```
        .anyRequest().hasRole("ACTUATOR")
```

```
        .and()
```

```
        .httpBasic();
```

```
    }
```

```
}
```

## **Customization**

- Customize endpoints to expose only necessary information
- Create custom health indicators for application-specific health checks

## **Metrics**

- Use micrometer for consistent metrics collection
- Choose appropriate metrics for your application's needs

## **Monitoring**

- Integrate with monitoring tools (e.g., Prometheus, Grafana)
- Set up alerts for critical metrics

## **Documentation**

- Document custom endpoints and metrics
- Provide clear guidelines for operations team

## **Performance**

- Be mindful of the performance impact of enabled endpoints
- Use sampling for high-volume metrics

## **Versioning**

- Include application version in the info endpoint
- Use git commit information for precise versioning

## **Healthchecks**

- Implement meaningful health checks for external dependencies
- Use the health endpoint for load balancer checks

## Sample Output After Visiting (<http://localhost:8080/actuator>)

```
{
  "_links": {
    "self": {
      "href": "http://localhost:8080/actuator",
      "templated": false
    },
    "beans": {
      "href": "http://localhost:8080/actuator/beans",
      "templated": false
    },
    "caches-cache": {
      "href": "http://localhost:8080/actuator/caches/{cache}",
      "templated": true
    },
    "caches": {
      "href": "http://localhost:8080/actuator/caches",
      "templated": false
    },
    "health": {
      "href": "http://localhost:8080/actuator/health",
      "templated": false
    },
    "health-path": {
      "href": "http://localhost:8080/actuator/health/{*path}",
      "templated": true
    },
    "info": {
      "href": "http://localhost:8080/actuator/info",
      "templated": false
    },
    "conditions": {
```

```
"href": "http://localhost:8080/actuator/conditions",
"templated": false
},
"configprops": {
  "href": "http://localhost:8080/actuator/configprops",
  "templated": false
},
"configprops-prefix": {
  "href": "http://localhost:8080/actuator/configprops/{prefix}",
  "templated": true
},
"env": {
  "href": "http://localhost:8080/actuator/env",
  "templated": false
},
"env-toMatch": {
  "href": "http://localhost:8080/actuator/env/{toMatch}",
  "templated": true
},
"loggers": {
  "href": "http://localhost:8080/actuator/loggers",
  "templated": false
},
"loggers-name": {
  "href": "http://localhost:8080/actuator/loggers/{name}",
  "templated": true
},
"heapdump": {
  "href": "http://localhost:8080/actuator/heapdump",
  "templated": false
},
"threaddump": {
```



```
    "href": "http://localhost:8080/actuator/threaddump",
    "templated": false
  },
  "metrics-requiredMetricName": {
    "href": "http://localhost:8080/actuator/metrics/{requiredMetricName}",
    "templated": true
  },
  "metrics": {
    "href": "http://localhost:8080/actuator/metrics",
    "templated": false
  },
  "sbom-id": {
    "href": "http://localhost:8080/actuator/sbom/{id}",
    "templated": true
  },
  "sbom": {
    "href": "http://localhost:8080/actuator/sbom",
    "templated": false
  },
  "scheduledtasks": {
    "href": "http://localhost:8080/actuator/scheduledtasks",
    "templated": false
  },
  "mappings": {
    "href": "http://localhost:8080/actuator/mappings",
    "templated": false
  }
}
```