

Monitoring

Spring Boot Actuator Setup

Production-Ready features

- Definition of Actuator
 - An actuator is a manufacturing term, referring to a mechanical device for moving or controlling something. Actuators can generate a large amount of motion from a small change.

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

Endpoints

Endpoints

- Monitoring the application and
 - Interact with it as well e.g. Loggers
- Many endpoints available depending on starters, custom endpoints possible
- Most endpoints are sensitive by default
- /actuator/health – Shows application health information
- /actuator/info – Displays arbitrary application info
- /actuator/metrics – Shows 'metrics' information
- /actuator/trace – Displays trace information
- /actuator/loggers – Shows logger details

Customizing Existing Endpoints

- Endpoint property format
 - endpoints.[name].[property]
- Three properties are available:
 - id – path to access over HTTP
 - enabled – accessible or not
 - sensitive – authorization, security context needed

```
endpoints.beans.id=springbeans #rename to /springbeans
endpoints.beans.sensitive=false #disable security
endpoints.beans.enabled=true #enabled
```

```
management.security.enabled=false
#disable security check for all endpoints
```

/actuator/health Endpoint

- Check the health/status of running application
- Can be used by monitoring software to alert
- Information collected from beans implementing HealthIndicator interface
- Sensitive by default, can be changed

/actuator/info Endpoint

- Customizing app information

```
info.app.name=Spring Demo Application  
info.app.description=My Spring Boot Demo application  
info.app.version=1.0.0
```

```
{  
  "app": {  
    "version": "1.0.0",  
    "description": "My Spring Boot Demo application",  
    "name": "Spring Demo Application"  
  }  
}
```


/actuator/metrics Endpoint

- Publishes information about OS, JVM and Application level metrics
 - memory, heap, processors, threads, classes loaded, classes unloaded, thread, HTTP metrics ...

```
{  
  "names": [  
    "jvm.buffer.memory.used",  
    "jvm.memory.used",  
    "jvm.buffer.count",  
    "http.server.requests",  
    "jvm.memory.committed",  
    "jvm.buffer.total.capacity",  
    "jvm.memory.max",  
    "cpu",  
    ...  
  ]  
}
```

/actuator/loggers Endpoint

- Application logging levels at runtime
- /application/loggers or /logger/{logger}

```
{  
  "levels": ["OFF", "ERROR", "WARN", "INFO", "DEBUG", "TRACE"],  
  "loggers": {  
    "ROOT": {  
      "configuredLevel": null,  
      "effectiveLevel": "TRACE"  
    },  
    ...  
  }  
}
```

/actuator/loggers Endpoint update

- Changing application logging levels at runtime
- /application/logger/{logger} partial update

```
$ curl -i -X POST -H 'Content-Type: application/json' \
-d '{"configuredLevel": "DEBUG"}' \
http://localhost:8080/actuator/loggers/ROOT
```

Application (Development) Info

/actuator/info with Build information (1)

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>build-info</goal>
      </goals>
      <configuration>
        <additionalProperties>
          <encoding.source>UTF-8</encoding.source>
          <encoding.reporting>UTF-8</encoding.reporting>
          <java.source>${maven.compiler.source}</java.source>
          <java.target>${maven.compiler.target}</java.target>
        </additionalProperties>
      </configuration>
    </execution>
  </executions>
</plugin>
```

/actuator/info with Build information (2)

```
{
  "app": {
    "version": "1.0",
    "description": "My Spring Boot Demo application",
    "name": "Spring Demo Application"
  },
  "build": {
    "name": "demo",
    "time": 1510728931000,
    "java": {
      "target": "1.8",
      "source": "1.8"
    },
    "encoding": {
      "source": "UTF-8",
      "reporting": "UTF-8"
    },
    "version": "0.0.1-SNAPSHOT",
    "group": "com.example",
    "artifact": "demo"
  }
}
```

/actuator/info with GIT information

- Adding Git plugin

```
<plugin>
  <groupId>pl.project13.maven</groupId>
  <artifactId>git-commit-id-plugin</artifactId>
</plugin>
```

- Build, restart and refresh /actuator/info

```
{
  "app": {
    "name": "Spring Demo Application",
    ...
  },
  "git": {
    "commit": {
      "time": 1510729626000,
      "id": "d2a5dfb"
    },
    "branch": "master"
  }, ... }
```

Further Customization (1)

- Actuator endpoints over a non-standard port
- Restrict access of endpoints over the network

```
management.server.port=8081  
management.server.address=127.0.0.1
```

- Most endpoints sentive by default

```
management.endpoints.web.expose=*  
management.endpoints.web.exclude=env
```

- When using Spring Security configure URLs

Further Customization (2)

```
public class ActuatorWebSecurityConfigurationAdapter extends
    WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
            .requestMatchers(EndpointRequest.to("status", "info"))
                .permitAll()
            .requestMatchers(EndpointRequest.toAnyEndpoint())
                .hasRole("ACTUATOR")
            .requestMatchers(StaticResourceRequest.toCommonLocations())
                .permitAll()
            .antMatchers("/**")
                .hasRole("USER")
            .and()
                .httpBasic();
    }
}
```