

Qué son los actuators de Spring Boot



Luis Fernandez

Feb 25, 2018 · 3 min read



Por resumirlo brevemente, los actuators de Spring Boot ofrecen funcionalidades listas para en entorno de producción.

Supervisan nuestra aplicación, recopilan métricas, comprenden y analizan el tráfico y el estado de nuestra base de datos, y todo ello sin falta de tener que implementarlo por nuestra cuenta.

Los Actuators se utilizan principalmente para exponer información operacional sobre nuestra aplicación en ejecución (*health, metrics, info, dump, env*, etc.)

¿Cómo se instala?

Lo primero es agregar la dependencia de Spring Boot Actuator de la siguiente manera.

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

Y actualizar nuestras dependencias Maven.

Endpoints

En la documentación oficial de Spring Boot existen muchos endpoints para los actuators que nos pueden ser de ayuda para nuestros entornos. Por resumir algunos de

los más importantes:

- */health*: Muestra información acerca de la salud de nuestra aplicación. Un simple “estado” si estamos sin autenticar o información mucho más detallada si estamso autenticados en la aplicación.
- */info*: Muestra información arbitraria de nuestra aplicación.
- */metrics*: Muestra la información de métricas de nuestra aplicación.
- */trace*: Muestra información de seguimiento (por defecto las últimas peticiones HTTP).

¿Cómo se configura?

Para la configuración de los endpoints se debe utilizar el fichero *application.properties* que nos provee Spring Boot con el siguiente formato:

endpoints.[endpoint name].[property to customize]

Tenemos tres propiedades disponibles:

- *id*: mediante el cual se accederá a este punto final por HTTP.
- *enabled*: si es true es accesible, en caso contrario, no.
- *sensitive*: si es true, necesitará que esté autorizado para mostrar información relevante acerca de HTTP.

Por ejemplo, en el caso de */health* tenemos varias opciones para mostrar la información de salud de nuestra aplicación.

Si queremos mostrar toda la información de la salud de nuestra aplicación sin falta de estar autenticado, utilizaremos la siguiente instrucción:

```
endpoints.health.sensitive=*  
management.security.enabled=false
```

Por el contrario, si queremos mostrar sólo el estado de la salud de nuestra aplicación sin estar autenticado, utilizaremos:

```
endpoints.health.sensitive=false  
management.security.enabled=true
```

Por último, si no queremos mostrar nada sin estar autenticado, utilizaremos:

```
endpoints.health.sensitive=true  
management.security.enabled=true
```

Por defecto, la información que nos aparece en el caso del endpoint */health* es:

```
{  
  "status" : "UP"  
}
```

Como vemos, esta información podría ser suficiente para saber el estado de nuestro sistema, en este caso que está levantado.

Pero podemos ver más información, en función de cómo configuremos las properties de los actuators, como por ejemplo:

```
{  
  "status" : "DOWN",  
  "myHealthCheck" : {  
    "status" : "DOWN",  
    "Error Code" : 1  
  },  
  "diskSpace" : {  
    "status" : "UP",  
    "free" : 209047318528,  
    "threshold" : 10485760  
  }  
}
```

En el caso de */info* podemos customizar la información que nos muestra de la siguiente manera:

```
info.app.name=Spring Sample Application  
info.app.description=This is my first spring boot application  
info.app.version=1.0.0
```

Y la salida sería:

```
{  
  "app" : {  
    "version" : "1.0.0",  
    "description" : "This is my first spring boot application",
```

```
"name" : "Spring Sample Application"
}
}
```

Para las métricas utilizaremos */metrics* que publica información sobre el sistema operativo, la JVM, y más tipos de métricas. Tendría la siguiente estructura:

```
{
"mem" : 193024,
"mem.free" : 87693,
"processors" : 4,
"instance.uptime" : 305027,
"uptime" : 307077,
"systemload.average" : 0.11,
"heap.committed" : 193024,
"heap.init" : 124928,
"heap.used" : 105330,
"heap" : 1764352,
"threads.peak" : 22,
"threads.daemon" : 19,
"threads" : 22,
"classes" : 5819,
"classes.loaded" : 5819,
"classes.unloaded" : 0,
"gc.ps_scavenge.count" : 7,
"gc.ps_scavenge.time" : 54,
"gc.ps_marksweep.count" : 1,
"gc.ps_marksweep.time" : 44,
"httpsessions.max" : -1,
"httpsessions.active" : 0,
"counter.status.200.root" : 1,
"gauge.response.root" : 37.0
}
```

Con esto vemos cómo configurar de manera sencilla para entornos preparados para producción los endpoints sin falta de tocar líneas de código ni implementar ninguna estructura compleja para ver la salud o la información de nuestra aplicación.

Se puede consulta toda la información de endpoints disponibles en la documentación oficial de Spring Boot por si necesitaramos alguno en especial. En este artículo se han visto los más utilizados pero existen muchos más.

[Actuator](#) [Spring Boot](#) [Spring](#) [Arquitectura](#)

[About](#) [Help](#) [Legal](#)