

## 2. Spring Boot – Actuators

### 📖 Case Study: Monitoring an Inventory System

#### Problem Statement:

You deploy an Inventory Management app and want to monitor its health, memory usage, bean loading, and environment settings without building these endpoints manually.

#### 📖 Key Concept:

Spring Boot Actuator exposes production-ready features like health checks, metrics, beans, and custom endpoints.

#### 📖 Scenario:

You add the spring-boot-starter-actuator dependency, and enable the / actuator endpoint in application.properties.

With zero code changes, you get:

- /actuator/health → Health of the service.
- /actuator/beans → Beans created in the container.
- /actuator/metrics → JVM and HTTP metrics.
- /actuator/env → Current environment values.

#### Solution:

Just add this line in application properties

# Expose only health and beans endpoint

management.endpoints.web.exposure.include=health,info,beans,metrics,env

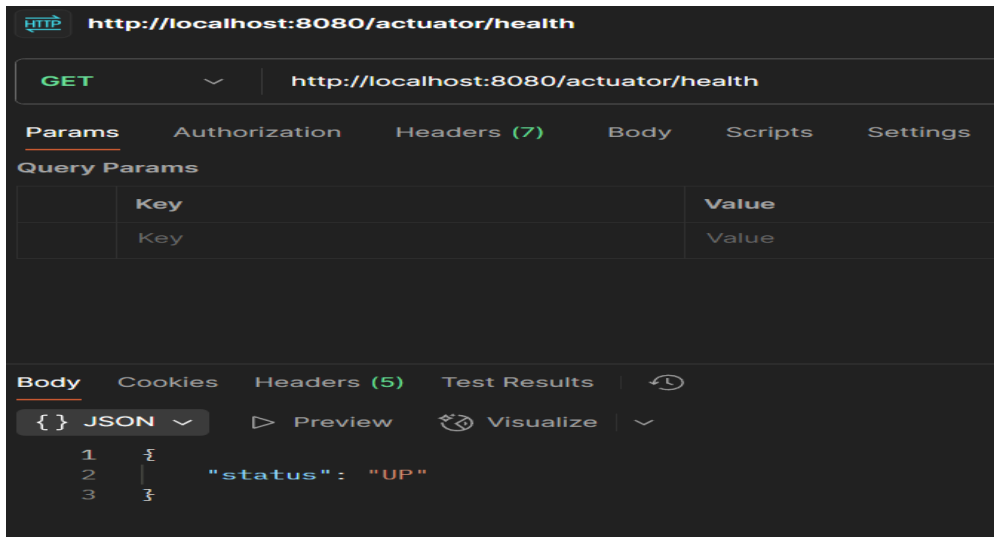
# Expose all actuator endpoints

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

#management.endpoints.web.base-path=/manage

And run java file

## SCREENSHOT OF OUTPUTS



HTTP **GET** <http://localhost:8080/actuator/health>

Params Authorization Headers (7) Body Scripts Settings

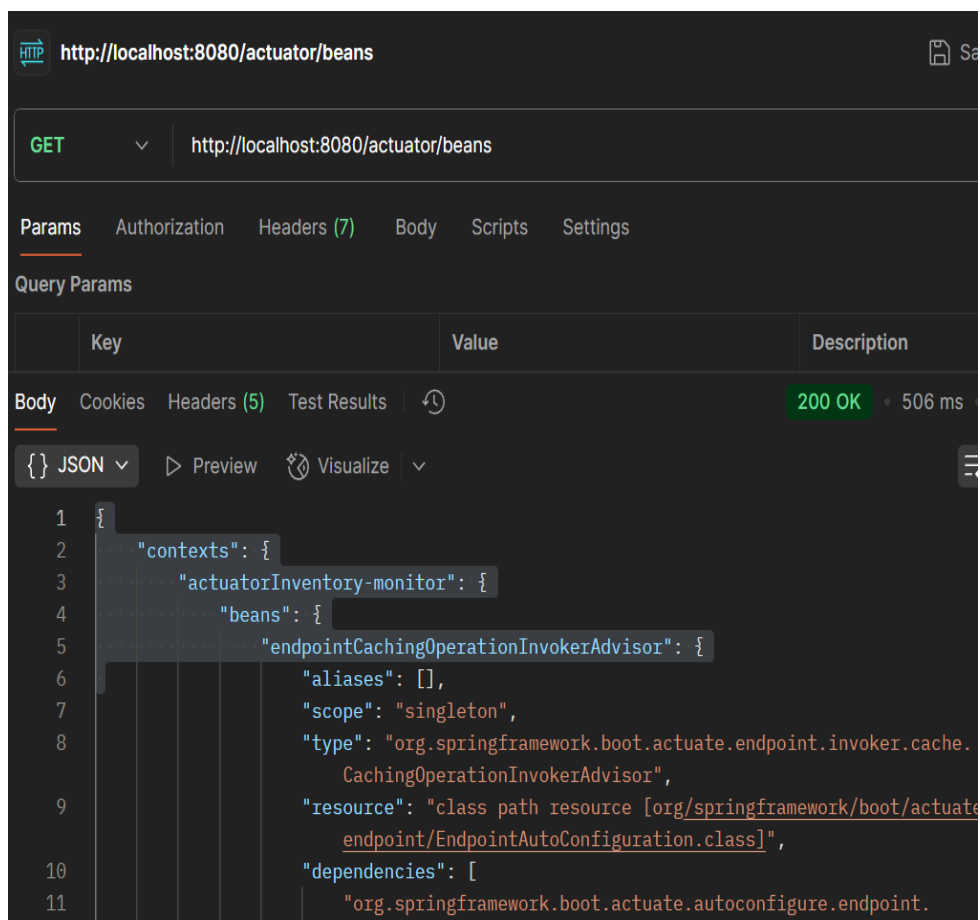
Query Params

Key	Value
Key	Value

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
1 {
2   "status": "UP"
3 }
```



HTTP **GET** <http://localhost:8080/actuator/beans>

Params Authorization Headers (7) Body Scripts Settings

Query Params

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (5) Test Results **200 OK** • 506 ms

{ } JSON Preview Visualize

```
1 {
2   "contexts": {
3     "actuatorInventory-monitor": {
4       "beans": {
5         "endpointCachingOperationInvokerAdvisor": {
6           "aliases": [],
7           "scope": "singleton",
8           "type": "org.springframework.boot.actuate.endpoint.invoker.cache.
9             CachingOperationInvokerAdvisor",
10          "resource": "class path resource [org/springframework/boot/actuate
11            endpoint/EndpointAutoConfiguration.class]",
12          "dependencies": [
13            "org.springframework.boot.actuate.autoconfigure.endpoint."
```

HTTP **http://localhost:8080/actuator/env**

GET **http://localhost:8080/actuator/env**

Params Authorization Headers (7) Body Scripts Settings


Query Params

	Key	Value	Description
--	-----	-------	-------------

Body Cookies Headers (5) Test Results ↻

{ } JSON ▾ ▶ Preview 🔄 Visualize ▾

```
1 {
2   "activeProfiles": [],
3   "defaultProfiles": [
4     "default"
5   ],
6   "propertySources": [
7     {
8       "name": "server.ports",
9       "properties": {
10        "local.server.port": {
11          "value": "*****"
12        }
13      }
14    },
```

HTTP **http://localhost:8080/actuator/metrics**  S

GET **http://localhost:8080/actuator/metrics**

Params Authorization Headers (7) Body Scripts Settings

Query Params

	Key	Value	Description
--	-----	-------	-------------

Body Cookies Headers (5) Test Results ↻ **200 OK** • 12 m

{ } JSON ▾ ▶ Preview 🔄 Visualize ▾

```
1 {
2   "names": [
3     "application.ready.time",
4     "application.started.time",
5     "disk.free",
6     "disk.total",
7     "executor.active",
8     "executor.completed",
9     "executor.pool.core",
10    "executor.pool.max",
11    "executor.pool.size",
12    "executor.queue.remaining",
```