**Spring Boot Actuator Module**

Spring boot’s module Actuator allows you to monitor and manage application usages in production environment, without coding and configuration for any of them. These monitoring and management information is exposed via [REST](https://restfulapi.net/) like endpoint URLs.

* 1. **Actuator Maven Dependency**

**To enable Spring Boot actuator endpoints to your Spring Boot application, we need to add the Spring Boot Starter actuator dependency in our build configuration file.**

|  |
| --- |
| POM.xml |
| **<dependency>**  **<groupId>org.springframework.boot</groupId>**  **<artifactId>spring-boot-starter-actuator</artifactId>**  **</dependency>**  By default, only **/health** and **/info** are exposed via Web APIs. Rest are exposed via JMX. Use management.endpoints.web.exposure.include=\* to expose all endpoints through the Web APIs.   |  | | --- | | **application.properties** | | **management.endpoints.web.exposure.include=\***    **# To expose only selected endpoints**  **#management.endpoints.jmx.exposure.include=health,info,env,beans** | |

The application will start on port 8080 by default. Once the application has started, you can list all the actuator endpoints exposed over HTTP from the URL <http://localhost:8080/actuator>

Let’s explore the health endpoint by opening the [**http://localhost:8080/actuator/health**](http://localhost:8080/actuator/health)

The status will be UP as long as the application is healthy. It will show DOWN if the application gets unhealthy due to any issue like connectivity with the database or lack of disk space etc.

Some of important and widely used actuator endpoints are given below:

|  |  |
| --- | --- |
| **ENDPOINT** | **USAGE** |
| /auditevents | Returns all auto-configuration candidates and the reason why they ‘were’ or ‘were not’ applied. |
| /beans | Returns a complete list of all the Spring beans in your application. |
| /mappings | Displays a collated list of all @RequestMapping paths.. |
| /env | Returns list of properties in current environment |
| /health | Returns application health information. |
| /caches | It exposes available caches. |
| /conditions | Shows the conditions that were evaluated on configuration and auto-configuration. |
| /configprops | It displays a collated list of all @ConfigurationProperties. |
| /integrationgraph | It shows the Spring Integration graph. Requires a dependency on spring-integration-core. |
| /loggers | The configuration of loggers in the application.. |
| /scheduledtasks | Displays the scheduled tasks in the application. |
| /sessions | Returns trace logs (by default the last 100 HTTP requests). Requires an HttpTraceRepository bean. |
| /httptrace | It allows retrieval and deletion of user sessions from a Spring Session-backed session store. Requires a Servlet-based web application using Spring Session. |
| /shutdown | Lets the application be gracefully shutdown. Disabled by default. |
| /threaddump | It performs a thread dump. |
| /metrics | It shows several useful metrics information like JVM memory used, system CPU usage, open files, and much more. |