

What is the benefit of spring boot actuators ?

=> These are non-functional features that require in any project's deployment and maintenance .. Earlier teams used dependent various tools given by jdk and third party to get these non-functional features.. now most of these operations can be done using actuators

=> For example when bug is raised we need logging info and thread dump info
Earlier people have used separate logging tools like ELK stack and etc..
also separate thread dump tools like jconsole, jmc, jstack, Visual VM and etc..
After the arrival of spring boot .. we can use actuators support like loggers , thread dump for same.

=> Earlier few non-functional features have been developed using Spring AOP by placing lots of efforts .. After the arrival of spring boot actuators we are getting more facilities as ready made facilities to deal with non-functional features

=> log4j, slf4j and etc are called logging tools

=> splunk, ELK stack and etc.. are called Log monitoring tools

To get all actuators

<http://localhost:9901/actuator>

List of Imp actuators

health, info, configprops, mappings, threaddump, heapdump, env, beans, refresh, loggers, scheduledtasks, cache, metrics and etc.. (nearly 15-actuators are there)

beans :: Gives info about all spring beans of current spring boot App (Both AutoConfigured and manually configured)

URL :: <http://localhost:9901/actuator/beans>

How to add Json Formatter Extension to chrome?

=> <https://chrome.google.com/webstore/detail/json-formatter/bcjndccogepfmagmhmhkgooihoe>
(open this url chrome browser -> add to chrome -> install ...)

configprops :: Gives all the @ConfigurationProperties based key-value loadings done both user-defined and pre-defined spring bean classes.

<http://localhost:9901/actuator/configprops>

env :: Gives the current env. of the Project in the form of key-value pairs like jvm files added \$classpath , cpu details , os details and etc..
<http://localhost:9901/actuator/env> (for all props)
<http://localhost:9901/actuator/env/user.dir> (specific single prop)

loggers :: Gives all log messages related details along with their logging levels.

<http://localhost:9901/actuator/loggers> trace=debug/info=warn/error=OFF (gives all packages info)

scheduledtasks :: Gives all @Scheduled methods info like InitialDelay , fixedDelay, fixedRate , cron and etc.. details

```

{
  "name": "com.example.demo",
  "properties": {
    "spring.profiles.active": "dev",
    "spring.datasource.url": "jdbc:mysql://localhost:3306/demo",
    "spring.datasource.username": "root",
    "spring.datasource.password": "root"
  },
  "actuator": {
    "enabled": true
  }
}
```

management.endpoint.configprops.show-values=always

(the configprops actuator displays all its values as the ** symbols, to replace them with real values, use the above key)

Now from Spring boot version 3, by default the value of all properties are hidden.

To display the value in json endpoint, below configuration needs to be added in application.properties file:

management.endpoint.configprops.show-values=always (avoids **** symbol while displaying certain properties values)

value values for the Property is as below

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

management.endpoint.configprops.show-values=always

What is the benefit of spring boot actuators ?

=====

=====

==

=> These are non-functional features that required in any project's deployment and maintenance .. Earlier teams used dependent various tools given by jdk

tools

and thirdparty to get these non-functional features.. now most of those operations can be done using actuators

=> For example when bug is raised we need logging info and threaddump info Earlier people have used separate logging tools like ELK stack and etc..

also separate thread dump tools like jconsole, JMC, jstack, Visual Vm and etc.. After the arrival of spring boot.. we can use actuators support like loggers, threaddump for same.

To get all actuators

=====

<http://localhost:9901/actuator>

List of imp actuators

=> Earlier few non-functional features have been developed using Spring AOP by placing lots efforts.. After the arrival of spring boot actuators we are getting more facilities as ready made facilities to deal with non-functional features

=> log4j, slf4j and etc are called logging tools

=> splunk, ELK stack and etc.. are called Log monitoring tools

health,info, configprops, mappings, threaddump, heapdump, env, beans, refresh loggers, scheduledtasks,cache,metrics and etc..

beans :: Gives info about all spring beans of current spring boot App (Both AutoConfigured and manually configured)

URL:: <http://localhost:9901/actuator/beans>

(nearly 15+actuators are there)

How to add Json Formatter Extension to chrome?

=> <https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa?hl=en>

(open this url chrome browser ---> add to chrome ---> install ...

configprops :: Gives all the @ConfigurationProperties based key=value loadings in done both user-defined and pre-fined spring bean classes.

<http://localhost:9901/actuator/configprops>

env :: Gives the current env.. of the Project in the form of key=value pairs like jars files added,classpath, cpu details, os details and etc.. <http://localhost:9901/actuator/env> (for all props)

<http://localhost:9901/actuator/env/user.dir> (specific single prop)

loggers :: Gives all log messages related details along with their logging levels.

management.endpoint.configprops.show-values-always

(the configprops actuator displays all its values as the *** symbols, To replace them with real values, use the above key)

Now from Spring boot version 3, by default the value of all properties are hidden.

To display the value in /env endpoint, below configuration needs to be added in application.properties file.

```
management.endpoint.env.show-values=ALWAYS
```

Valid values for the Property is as below

trace<debug<info<warn<error<OFF <http://localhost:9901/actuator/loggers> (gives all packages info)
<http://localhost:9901/actuator/com.nt> (give only com.nt package info) **scheduledtasks** : Gives all @Scheduled methods info like initialDelay, fixed Delay, fixed Rate,

cron and etc.. details

← → C

localhost:9901/actuator/scheduledtasks

```
"cron": [], "fixedDelay": [
```

```
▼ "runnable": {
```

ALWAYS

NEVER (default)

WHEN AUTHORIZED

(avoids ***** symbol while displaying certain properties values)

threaddump ::

```
=====
```

```
"target": "com.nt.scheduling. SampleTask.testers"
```

```
},
```

```
"initialDelay": 0,
```

```
"interval": 10000
```

```
}
```

```
],
```

```
"fixedRate": [],
```

```
"custom": []
```

How to know wheather certain actuator allows the param or not?

Ans) when we list all the actuators.. we can see the actuator param details

Gives info about all the daemon threads(backgrounds) that are created and running continusoly

<http://localhost:9901/actuator/threaddump>

For Example

```
▼ "configprops":{
```

```
},
```

```
"href": "http://localhost:6061/actuator/configprops",
```

```
"templated": false
```

```
▼ "configprops-prefix": {  
  "href": "http://localhost:6061/actuator/configprops/{prefix}",  
  "templated": true
```

specifies that this actuator allows me take the parameter

about

mappings :: gives info all handler methods, RestController methods mappings info

=====

like request path, method name, method signature, method type and etc.. (Basically it gives all endpoint details) <http://localhost:9901/actuator/mappings>

of

able

heapdump :: Gives gives heapdump (memory details) in the form download byte code info.. So we need additional tools to analyze that content like Eclipse Memory Analyzer, IBMheapMemoryAnalyzer, heap hero and etc...

<http://localhost:9901/actuator/heapdump>

To analyze the heap dump

Add this additional property form spring boot 3.5 onwards #enable heapdump

management.endpoint.heapdump.access=unrestricted

gives "heapdump" file in downloads folder having byte /machine code

go to

<https://heaphero.io/> select the above heapdump file -->

get the analyzation report

Total Size: 14.03mb

Memory wasted: 3.55mb (25%)

Duplicate Primitive Arrays 1.23mb

(8.8%)

</>

Class Count: 8,616

about

Object Count: 321,191

Thread Count : 24

Inefficient Collections 1.3mb (9.3%) -

Inefficient Object Arrays

575.31kb (4.0%)

Inefficient Primitive

Arrays 256.23kb (1.8%)

Duplicate Objects 183.94kb (1.3%)

Objects waiting finalization 48b (<0.1%)

Duplicate Strings n/a (0.0%)

caches :: gives info all CacheManagers that are configured in different parts of the application.

=====

http://localhost:9901/actuator/caches

metrics :: gives Info about properties list related to current project.

http://localhost:9901/actuator/metrics

localhost:9900/actuate X

← C

{

"name": "system.cpu.count",

Leak Suspect ☼

► spring boot logging by X ✓ New Tab localhost:9900/actuator/metrics/system.cpu.count

You are screen sharing

To get specific metric information

"description": "The number of processors available to the Java virtual machine",

"measurements": [

"statistic": "VALUE",

What is the base path for actuators?

http://localhost:9901/actuator

Can we change the basepath of actuator?

"value": 12

],

"availableTags": []

}

=>Like changing context path of web application.. we can also

change context path of actuators

in application.properties

To change basepath of actuators

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

To test the change

http://localhost:9901/nitinfo

http://localhost:9901/nitinfo/health http://localhost:9901/nitinfo/beans

note:: we can not take "/" as the basepath for actuators becoz it is already assigned to DispatcherServlet.

Q) While working with "refresh" actuator if External Config file (GitLab account file) and

local application.properties file of the MS project contains same keys with different values

Can u tell me which value will be injected?

GitLab's application.properties (External Config file)

dbuser=system

local application.properties (native config file)

dbuser=ramesh

In our MS

@Value("\${dbuser}")

private String user; --- gets what value ::

system

(Collects from external config file)

Memory leak occurs when objects are no longer being used by the application, but the Garbage Collector is unable to remove them from working memory because they're still being referenced.