

# Meeting non-functional requirements with Spring Boot Actuator

Vedran Pavić, Kapsch CarrierCom d.o.o.

## About the Author

- Software Development Engineer at Kapsch CarrierCom d.o.o. since 2011
- Developing solutions in Telco industry, primarily related to Number Portability
- An active contributor in open-source community (mostly Spring related projects)

## Non-functional requirement vs Actuator

- *In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.*  
([https://en.wikipedia.org/wiki/Non-functional\\_requirement](https://en.wikipedia.org/wiki/Non-functional_requirement))
- *An actuator is a type of motor that is responsible for moving or controlling a mechanism or system.*  
(<https://en.wikipedia.org/wiki/Actuator>)

## Enter Spring Boot Actuator

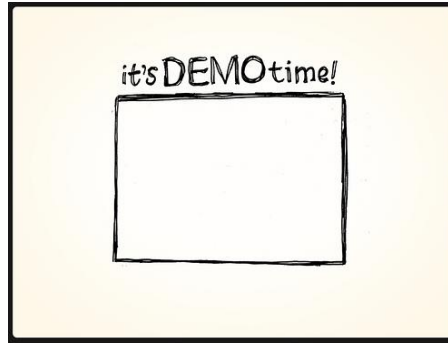
- Builds on Spring Boot foundations to provide production-ready features
- Focus on monitoring and management over HTTP, but supports other protocols as well
- Highly customizable and extendable, embraces other technologies

## Endpoints

- Enable monitoring capabilities for your applications (primarily over HTTP)
- Wide range of endpoint available out of the box: health information, application metrics, general application information, thread dump, environment information, trace information... plus many others
- Customizable via application properties
- Optional hypermedia support and endpoint browser

## Implementing Custom Endpoint

- Acuator endpoints are @Beans that implement Endpoint interface (or more specialized MvcEndpoint)
- Easily implement your own using AbstractEndpoint and AbstractEndpointMvcAdapter classes

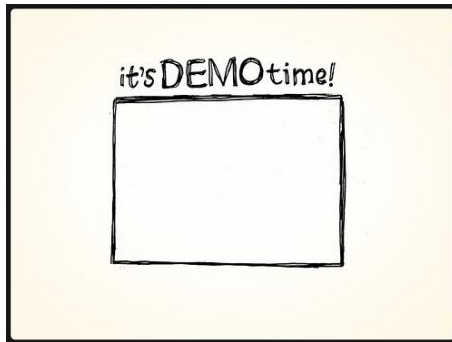


## Health Information

- Check the status of your application – useful for monitoring software, load-balancers, etc.
- Overall health information is contributed by multiple health indicators
- Many health indicators available out of the box, depending on what you use in your application (JDBC data source, JMS broker, Mail server...)

## Implementing Custom HealthIndicator

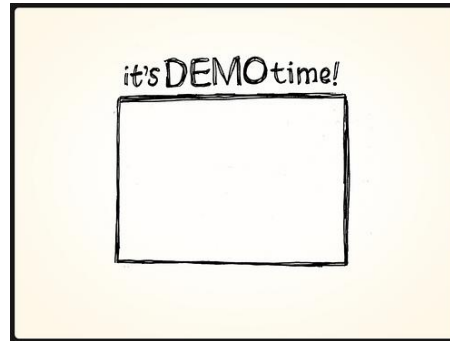
- Actuator health indicators are @Beans that implement HealthIndicator interface
- Easily implement your own using AbstractHealthIndicator class (also see CompositeHealthIndicator and HealthAggregator)





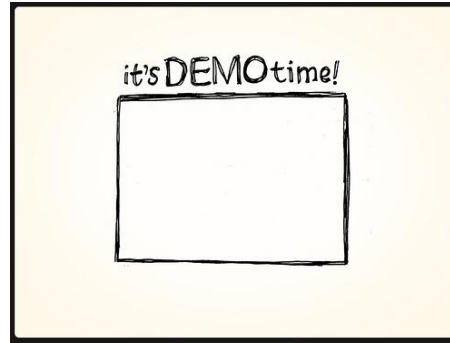
## Application Information

- Exposes various application information
- Information is collected from InfoContributors – environment/git/build contributors are provided



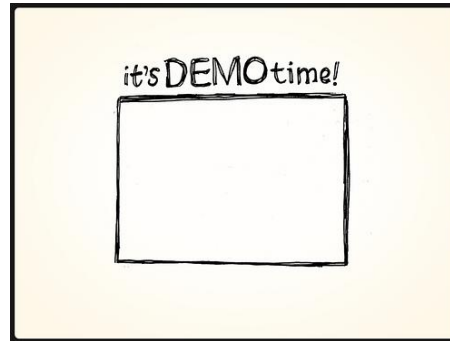
## JMX

- Acuator endpoint are also available over JMX
- JMX operations are available over HTTP using Jolokia



## Remote shell

- Monitoring is also possible via remote shell access (SSH, Telnet) using CRaSH
- Wide range of commands and utilities available out of the box – providing your own commands is easy



## Metrics

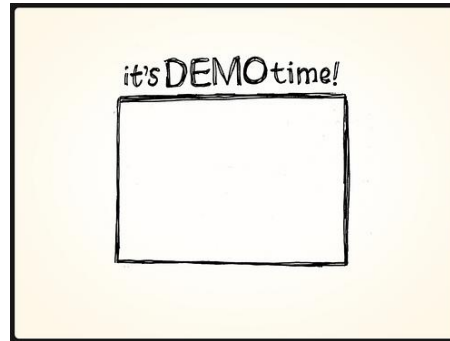
- Actuator automatically records system and HTTP metrics and exposes them using endpoint
- CounterService and GaugeService are available to record your own metrics – can be exposed using PublicMetrics @Beans
- Pluggable strategies for metric export and aggregation

## Audit Events

- Infrastructure for auditing – Spring Security’s authentication and authorization events are translated to Actuator’s AuditEvents
- AuditEventRepository implementation is used for storing and retrieval of events
- Simple to use for your own AuditEvents

## Bonus

- Spring Boot's build plugins allow creating fully executable JARs - can be installed as a system service
- Can be used to install application as a init.d or systemd service



## Resources

- Project page: <http://projects.spring.io/spring-boot/>
- Issue tracker: <https://github.com/spring-projects/spring-boot/issues>
- Source code: <https://github.com/spring-projects/spring-boot>
- Sample project: <https://github.com/vpavic/javacro16-spring-boot-actuator>

## Questions





**Vedran Pavić**

[vedran.pavic@kapsch.net](mailto:vedran.pavic@kapsch.net)

<https://www.kapsch.net/hr/kcc>

<https://github.com/vpavic>