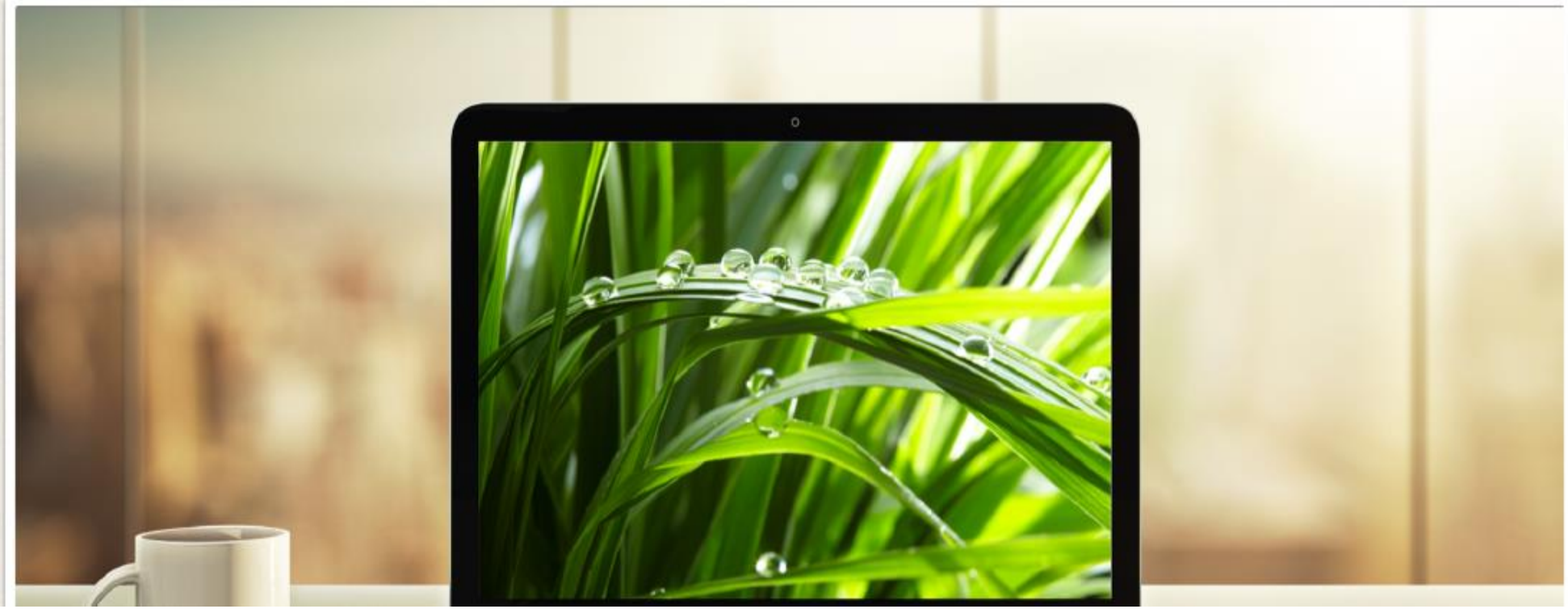
Spring Boot Actuator



Problem

- How can I monitor and manage my application?
- How can I check the application health?
- How can I access application metrics?

Solution: Spring Boot Actuator

- Exposes endpoints to monitor and manage your application
- You easily get DevOps functionality out-of-the-box
- Simply add the dependency to your POM file
- REST endpoints are automatically added to your application

No need to write additional code!

You get new REST endpoints for FREE!

Spring Boot Actuator

- Adding the dependency to your POM file

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

Spring Boot Actuator

- Automatically exposes endpoints for metrics out-of-the-box
- Endpoints are prefixed with: `/actuator`

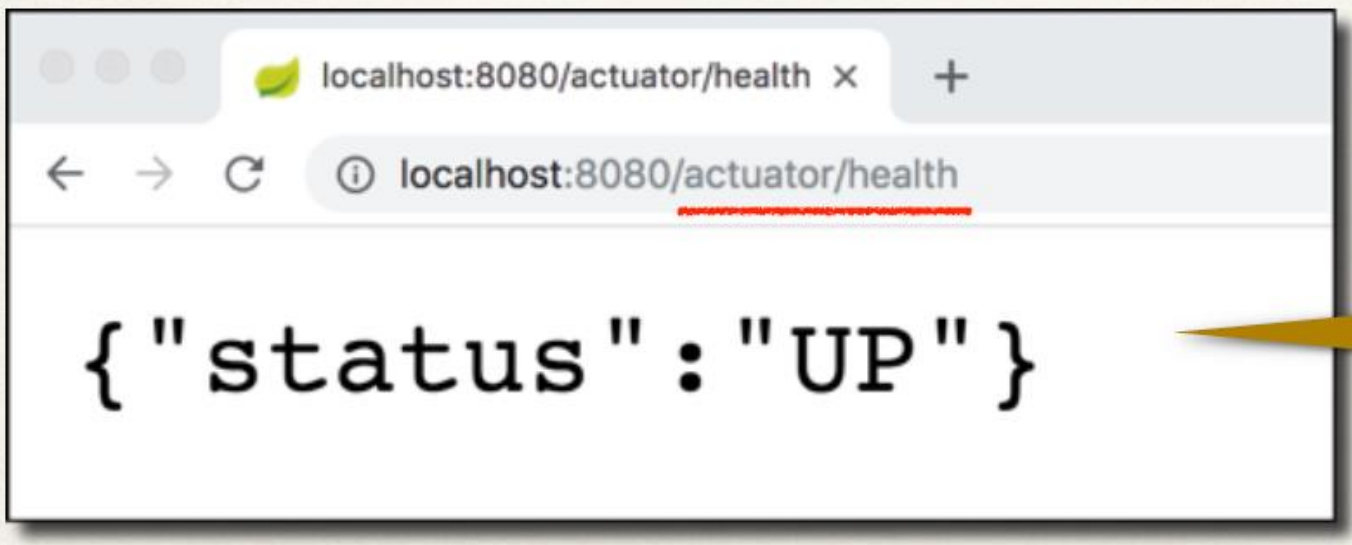
Name	Description
`/health`	Health information about your application
`/info`	Information about your project
...	

Remember:

You get these new REST endpoints for FREE!

Health Endpoint

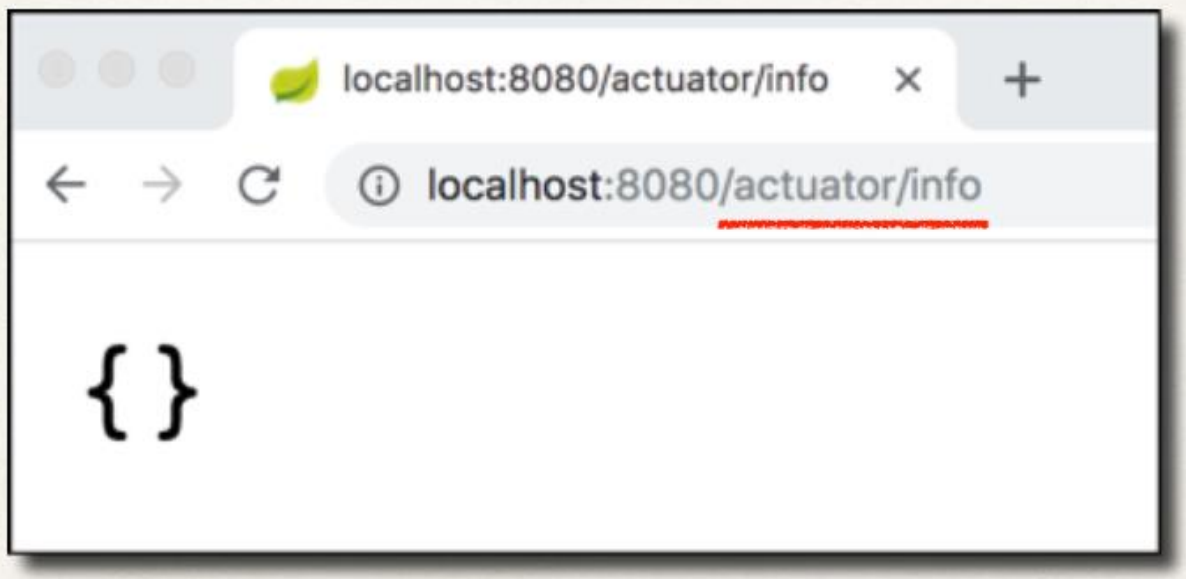
- **`/health`** checks the status of your application
- Normally used by monitoring apps to see if your app is up or down



Health status is customizable based on your own business logic

Info Endpoint

- `/info` gives information about your application
- Default is empty



Info Endpoint

- Update **`application.properties`** with your app info

File: src/main/resources/application.properties

```
info.app.name=My Super Cool App
info.app.description=A crazy and fun app, yoohoo!
info.app.version=1.0.0
```

Spring Boot Actuator Endpoints

- There are 10+ Spring Boot Actuator endpoints

Name	Description
`/auditevents`	**Audit events for your application**
`/beans`	**List of all beans registered in the Spring application context**
`/mappings`	**List of all @RequestMapping paths**
…	

Spring Boot Actuator Endpoints

Full list

`https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#production-ready-endpoints`

ID	Description
`auditevents`	Exposes audit events information for the current applic
`beans`	Displays a complete list of all the Spring beans in your
`caches`	Exposes available caches.
`conditions`	Shows the conditions that were evaluated on configura or did not match.
`configprops`	Displays a collated list of all `@ConfigurationPrope`
`env`	Exposes properties from Spring's `ConfigurableEnv`

Exposing Endpoints

- By default, only **`/health`** and **`/info`** are exposed
- To expose all actuator endpoints over HTTP

File: src/main/resources/application.properties

```
# Use wildcard "*" to expose all endpoints
# Can also expose individual endpoints with a comma-delimited list
#
management.endpoints.web.exposure.include=*
```

Get A List of Beans

- Access `http://localhost:8080/actuator/beans`


```
{
  "contexts": {
    "application": {
      "beans": {
        "endpointCachingOperationInvokerAdvisor": {
          "aliases": [],
          "scope": "singleton",
          "type": "org.springframework.boot.actuate.endpoint.invoker.cache.Cach
          "resource": "class path resource [org/springframework/boot/actuate/au
          "dependencies": [
            "environment"
          ]
        },
        "defaultServletHandlerMapping": {
          "aliases": [],
          "scope": "singleton",
          "type": "org.springframework.web.servlet.HandlerMapping",
          "resource": "class path resource [org/springframework/boot/autoconfig
          "dependencies": []
        },
        "org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfigura
          "aliases": [],
          "scope": "singleton",
          "type":
```


What about security??

We'll add security in later videos

Development Process

1. Edit **`pom.xml`** and add **`spring-boot-starter-acuator`**

2. View actuator endpoints for: **`/health`** and **`/info`**

3. Edit **`application.properties`** to customize **`/info`**