

Spring Boot Actuator



Problem

Problem

- How can I monitor and manage my application?

Problem

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

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

Solution: Spring Boot Actuator

- Exposes endpoints to monitor and manage your application

Solution: Spring Boot Actuator

- Exposes endpoints to monitor and manage your application
- You easily get DevOps functionality out-of-the-box

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

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

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

Spring Boot Actuator

- Adding the dependency to your POM file

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

Spring Boot Actuator

- Automatically exposes endpoints for metrics out-of-the-box

Spring Boot Actuator

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

Name	Description

Spring Boot Actuator

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

Name	Description
<code>/health</code>	Health information about your application

Spring Boot Actuator

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

Name	Description
<code>/health</code>	Health information about your application
<code>/info</code>	Information about your project

Spring Boot Actuator

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

Name	Description
<code>/health</code>	Health information about your application
<code>/info</code>	Information about your project
...	

Recent Updates in Spring Boot 2.6

19 November 2021

/info endpoint is not exposed by default.

You have to manually expose it and also enable custom application info properties.

Here are the steps:

- 1. Edit application.properties**
- 2. Add following entries**

```
management.endpoints.web.exposure.include=health,info  
management.info.env.enabled=true
```

Health Endpoint

Health Endpoint

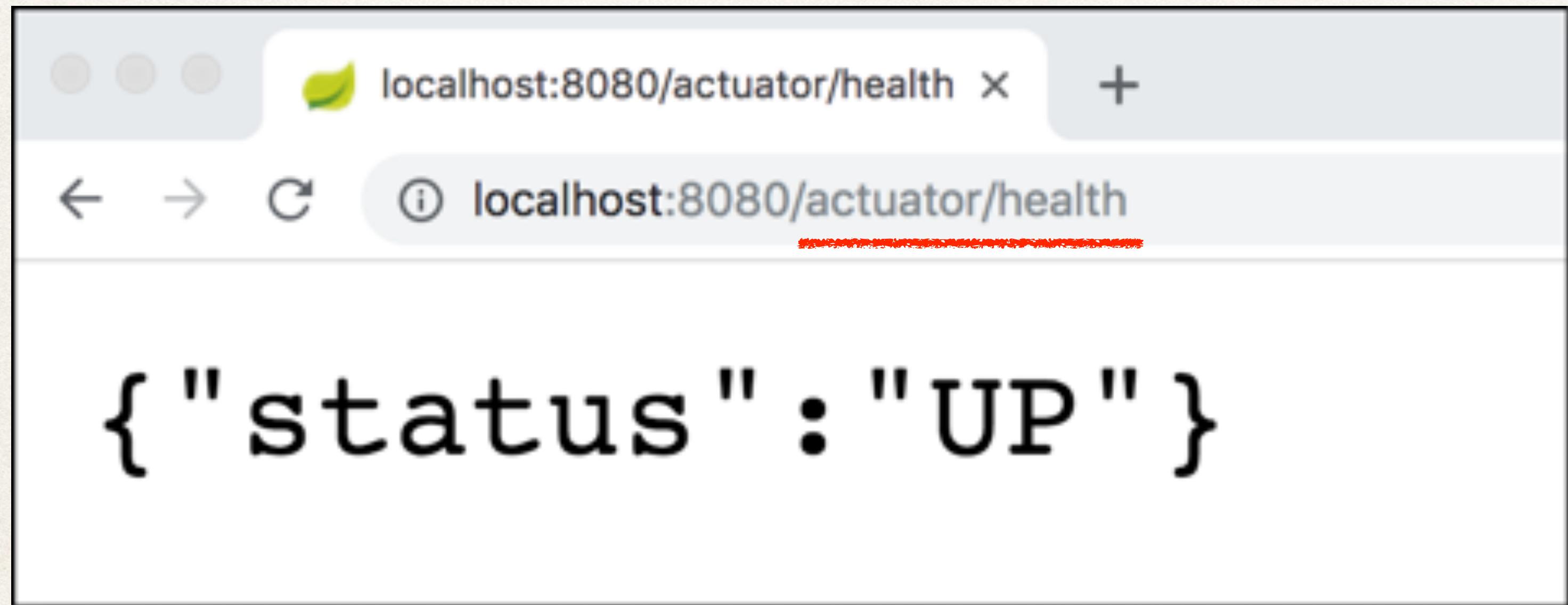
- **/health** checks the status of your application

Health Endpoint

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

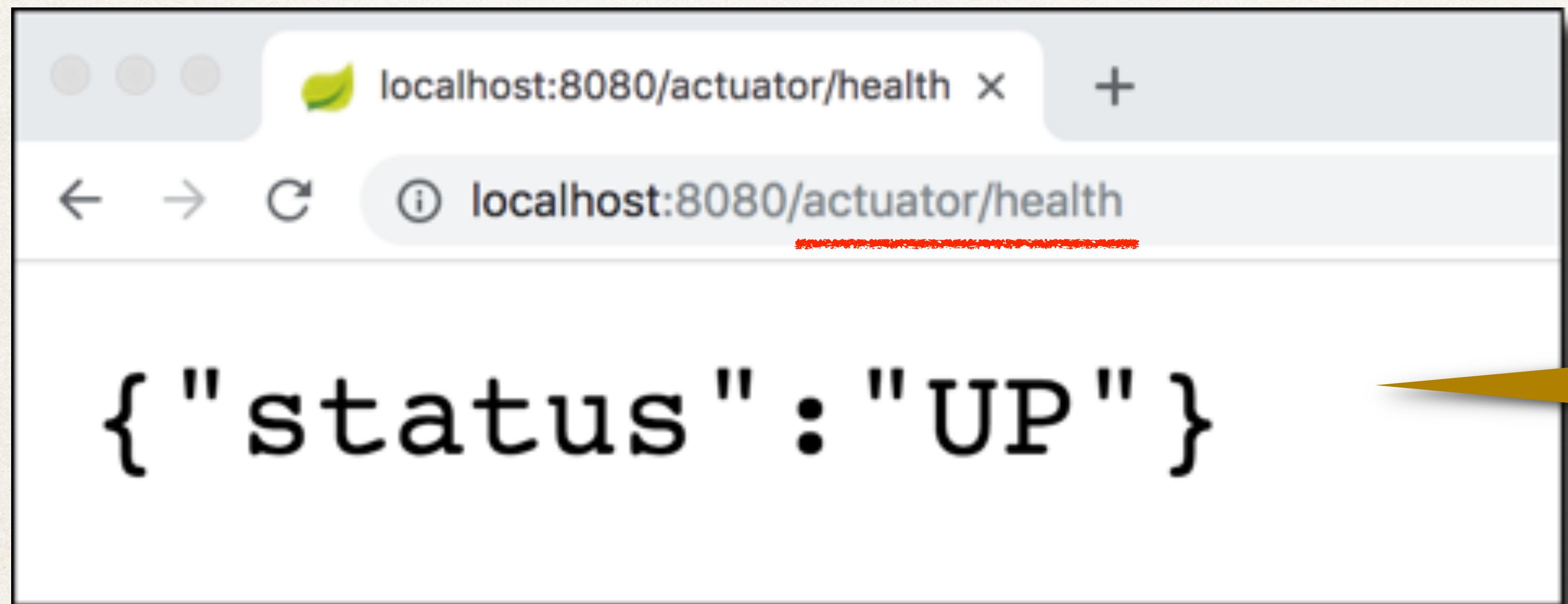
Health Endpoint

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



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

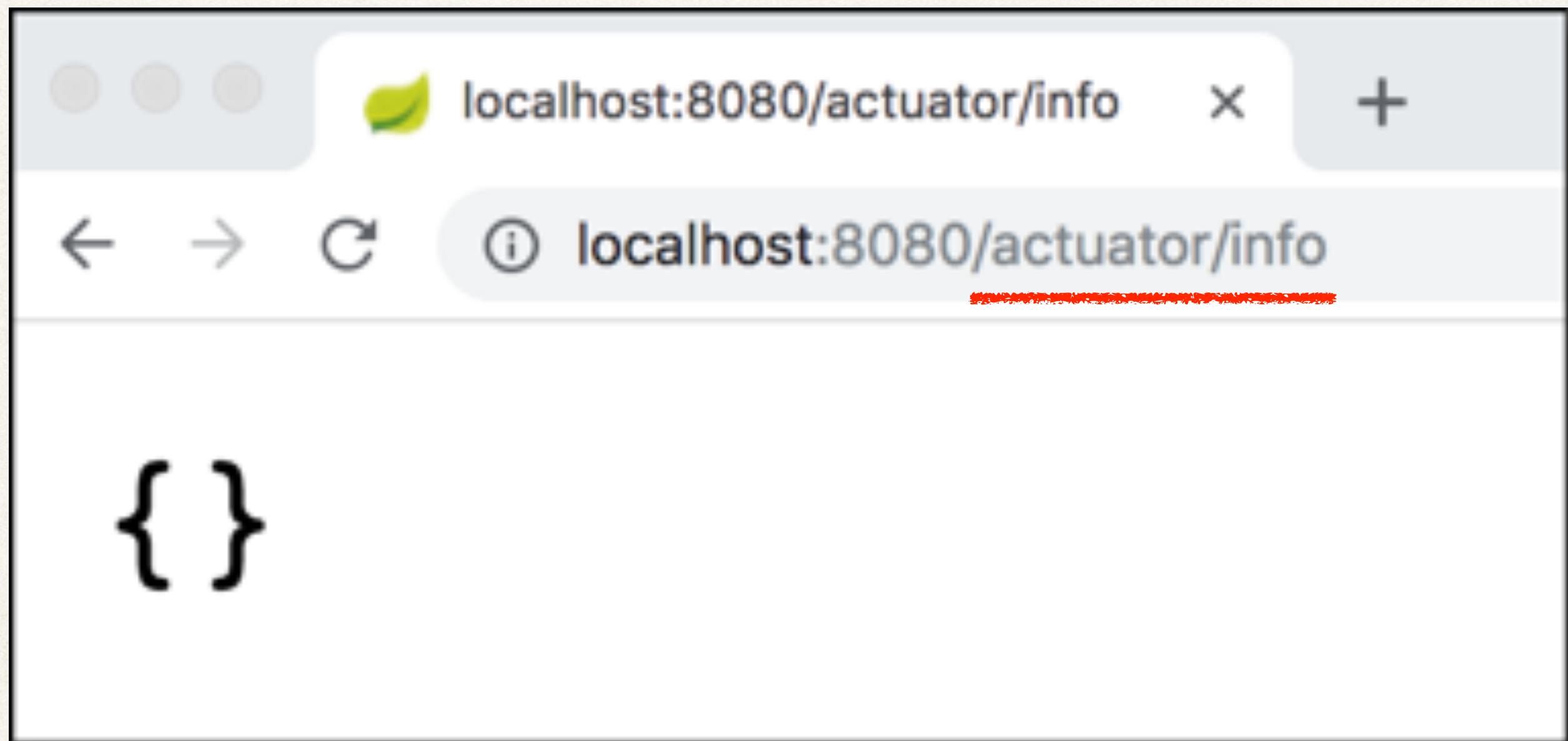
Info Endpoint

Info Endpoint

- `/info` gives information about your application

Info Endpoint

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



Info Endpoint

Info Endpoint

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

Info Endpoint

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

File: src/main/resources/application.properties

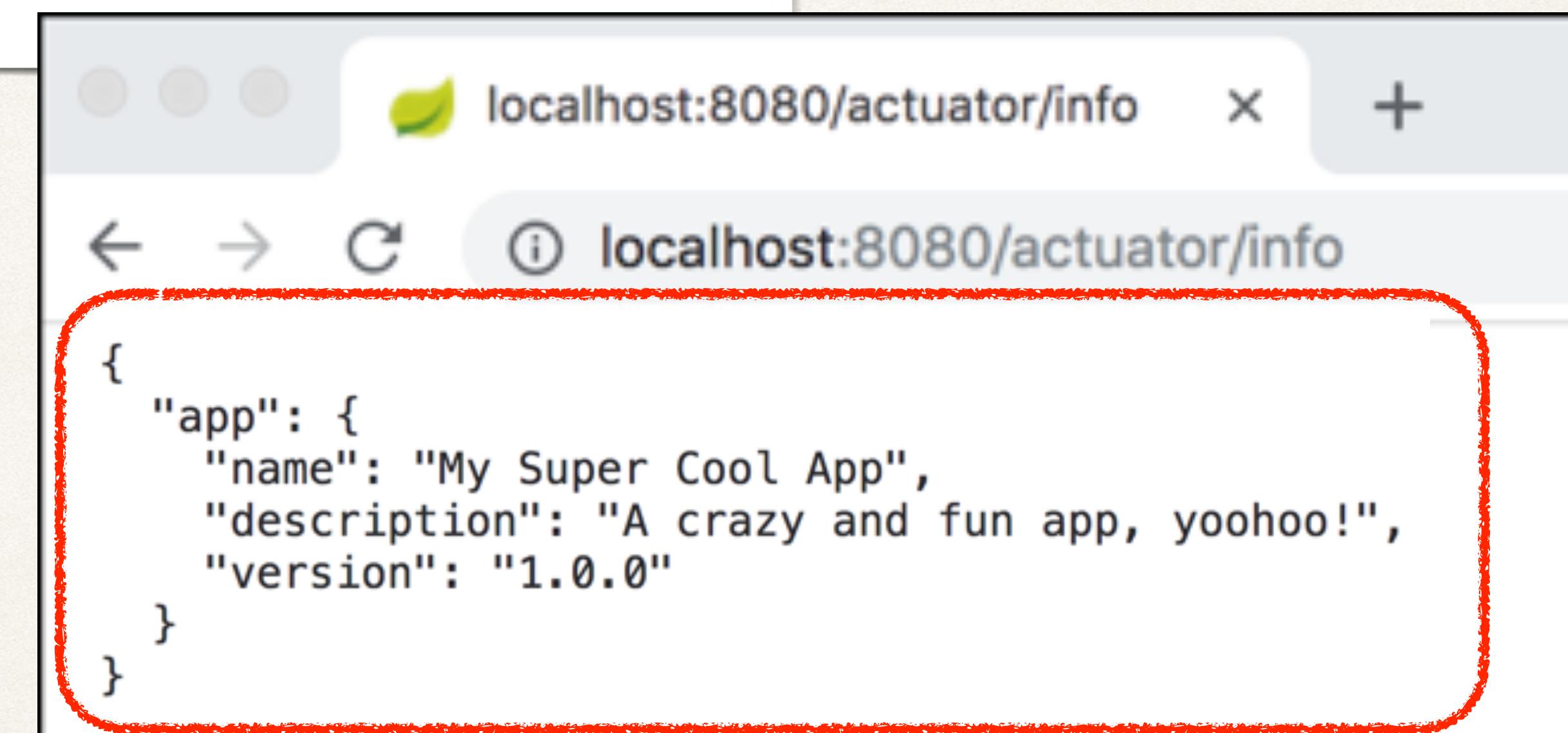
```
info.app.name=My Super Cool App
info.app.description=A crazy and fun app, yoohoo!
info.app.version=1.0.0
```

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

Spring Boot Actuator Endpoints

- There are 10+ Spring Boot Actuator endpoints

Name	Description

Spring Boot Actuator Endpoints

- There are 10+ Spring Boot Actuator endpoints

Name	Description
<code>/auditevents</code>	Audit events for your application

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

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

Spring Boot Actuator Endpoints

Full list

www.luv2code.com/actuator-endpoints

ID	Description
<code>auditevents</code>	Exposes audit events information for the current application.
<code>beans</code>	Displays a complete list of all the Spring beans in your application.
<code>caches</code>	Exposes available caches.
<code>conditions</code>	Shows the conditions that were evaluated on configuration properties and which did or did not match.
<code>configprops</code>	Displays a collated list of all <code>@ConfigurationProperties</code> annotated properties.
<code>env</code>	Exposes properties from Spring's <code>ConfigurableEnvironment</code> .

Exposing Endpoints

Exposing Endpoints

- By default, only **/health** and **/info** are exposed

Exposing Endpoints

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

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

Get A List of Beans

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

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":  
        }  
      }  
    }  
  }  
}
```

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

Step-By-Step

Development Process

Step-By-Step

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

Development Process

Step-By-Step

1. Edit **pom.xml** and add **spring-boot-starter-actuator**
2. View actuator endpoints for: **/health** and **/info**

Development Process

Step-By-Step

1. Edit **pom.xml** and add **spring-boot-starter-actuator**
2. View actuator endpoints for: **/health** and **/info**
3. Edit **application.properties** to customize **/info**