

A background image of the Golden Gate Bridge in San Francisco, viewed from a low angle looking down the length of the bridge towards the foggy horizon. The bridge's suspension cables and towers are visible, and the water below is calm.

Pivotal

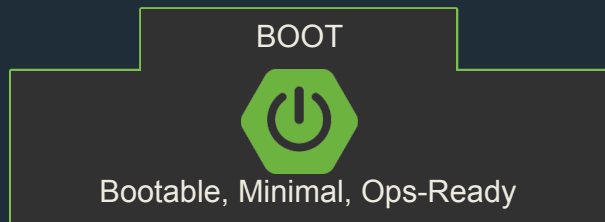
Cloud Native Applications

Advancing Spring Boot with Actuator and Profiles

Getting Started

Actuator

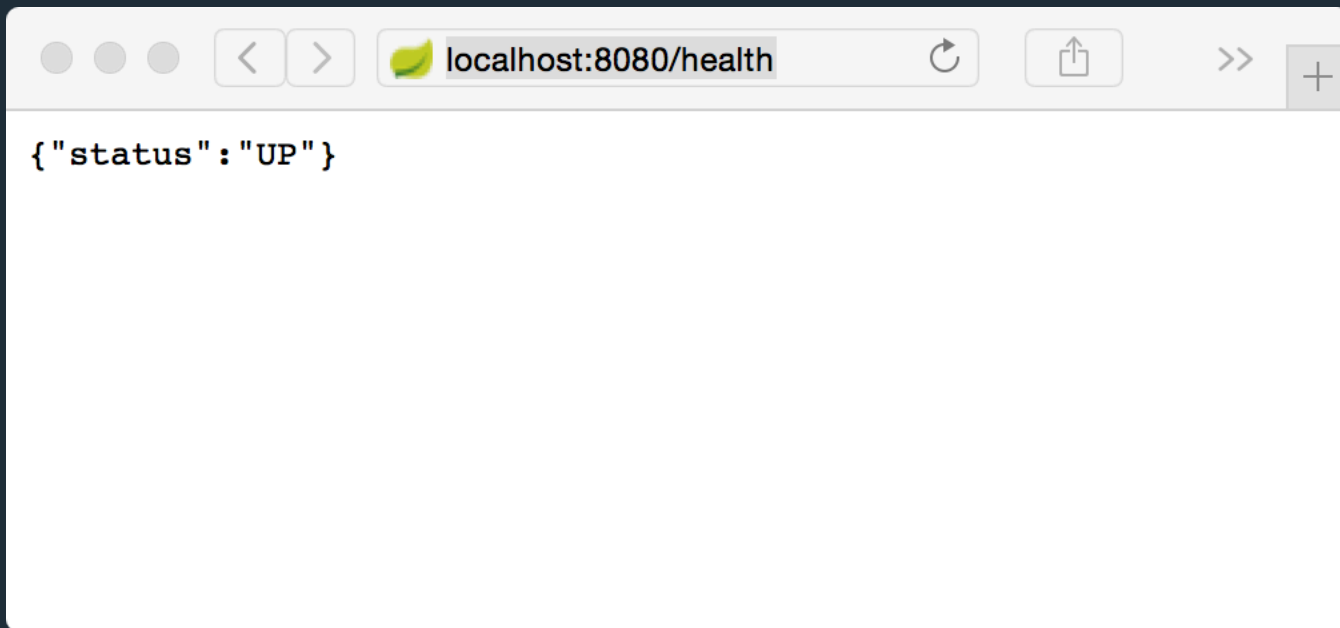
- Production grade features exposed as endpoints
 - Metrics
 - Health
 - Configuration
 - Errors
 - Info



Endpoints

ID	Description	Sensitive Default
<code>actuator</code>	Provides a hypermedia-based "discovery page" for the other endpoints. Requires Spring HATEOAS to be on the classpath.	true
<code>autoconfig</code>	Displays an auto-configuration report showing all auto-configuration candidates and the reason why they 'were' or 'were not' applied.	true
<code>beans</code>	Displays a complete list of all the Spring beans in your application.	true
<code>configprops</code>	Displays a collated list of all <code>@ConfigurationProperties</code> .	true
<code>docs</code>	Displays documentation, including example requests and responses, for the Actuator's endpoints. Requires <code>spring-boot-actuator-docs</code> to be on the classpath.	false
<code>dump</code>	Performs a thread dump.	true
<code>env</code>	Exposes properties from Spring's <code>ConfigurableEnvironment</code> .	true
<code>flyway</code>	Shows any Flyway database migrations that have been applied.	true
<code>health</code>	Shows application health information (when the application is secure, a simple 'status' when accessed over an unauthenticated connection or full message details when authenticated).	false
<code>info</code>	Displays arbitrary application info.	false

Health Endpoint



Health Indicators

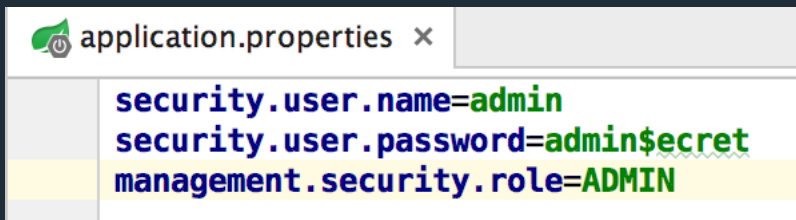
Health Indicator	Performed Checks
DiskSpaceHealthIndicator	Checks for low Disk space
DataSourceHealthIndicator	Check DataSource connection
ElasticsearchHealthIndicator	Checks ElasticSearch cluster is up
JmsHealthIndicator	Checks that a JMS broker is up
MailHealthIndicator	Checks that a mail server is up
MongoHealthIndicator	Checks that a Mongo database is up
RabbitHealthIndicator	Checks that a Rabbit server is up
RedisHealthIndicator	Checks that a Redis server is up
SolrHealthIndicator	Checks that a Solr server is up

HTTP Access

Importing Spring Security Dependency

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

Setting required Authentication & Authorization Roles



The screenshot shows an IDE window titled 'application.properties' with a close button. The file contains three lines of configuration: 'security.user.name=admin', 'security.user.password=admin\$secret', and 'management.security.role=ADMIN'. The text is color-coded: 'security' is blue, 'user' is blue, 'name=' is blue, 'admin' is green; 'password=' is blue, 'admin\$secret' is green; 'management' is blue, 'security' is blue, 'role=' is blue, 'ADMIN' is green.

```
security.user.name=admin  
security.user.password=admin$secret  
management.security.role=ADMIN
```

Metrics

Spring Boot Actuator includes a metrics service with 'gauge' and 'counter' support

Record your own metrics, inject CounterService and/or GaugeService into your bean

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.actuate.metrics.CounterService;
import org.springframework.stereotype.Service;

@Service
public class MyService {

    private final CounterService counterService;

    @Autowired
    public MyService(CounterService counterService) {
        this.counterService = counterService;
    }

    public void exampleMethod() {
        this.counterService.increment("services.system.myservice.invoked");
    }
}
```

```
{
  "mem":461824,
  "mem.free":206032,
  "processors":8,
  "instance.uptime":28635,
  "uptime":32978,
  "systemload.average":3.1455078125,
  "heap.committed":461824,
  "heap.init":262144,
  "heap.used":255791,
  "heap":3728384,
  "threads.peak":16,
  "threads.daemon":13,
  "threads":15,
  "classes":7717,
  "classes.loaded":7717,
  "classes.unloaded":0,
  "gc.ps_scavenge.count":7,
  "gc.ps_scavenge.time":50,
  "gc.ps_marksweep.count":2,
  "gc.ps_marksweep.time":80,
  "httpsessions.max":-1,
  "httpsessions.active":0,
  "datasource.primary.active":0,
  "datasource.primary.usage":0.0,
  "counter.status.200.metrics":2,
  "gauge.response.metrics":2.0
}
```

Profiles

- YAML file can contain for several documents
- Convenient to specify alternate configurations in the same file
- Set the active profile

SPRING_PROFILES_ACTIVE=production

or

spring.profiles.active=production

```
server:  
  port: 9000
```

```
spring:  
  profiles: development
```

```
server:  
  port: 9001
```

```
spring:  
  profiles: production
```

```
server:  
  port: 0
```

Development profile

Production profile

A dark, high-contrast photograph of a modern interior space, likely a transit hub or a large office building. The scene is filled with the silhouettes of many people walking or standing. Large windows in the background let in bright light, creating a strong backlight effect. A large clock is visible in the upper left corner. The floor is highly reflective, mirroring the silhouettes and the light from the windows. The overall mood is busy and architectural.

LAB