

# Spring Boot Actuator



# Spring Boot Actuator Nedir?

- Spring Boot, uygulamamızı çalışma zamanında **yönetmek ve monitör etmek** için bir takım kabiliyetler sunar
- Bu kabiliyetler **actuator** olarak adlandırılır

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

# Spring Boot Actuator Neler Sunar?

- Uygulama ile ilgili bir takım **konfigürasyon ayarlarına** erişme ve değiştirme imkanı sunar
- Çalışma zamanında **log konfigürasyonu** **inceleme ve değiştirme** imkanı sunar
- Uygulama ve sistem ile ilgili **pek çok metrik bilgisini öğrenme** imkanı sunar
  - Yüklenen sınıflar, heap size, thread count, veritabanı/cache erişimi, process sayısı, disk kapasitesi gibi

# Actuator Servislerine Nasıl Erişilir?

- Actuator endpoint'lerine erişmek için **HTTP endpoint'leri**, **JMX** veya **SSH/Telnet** gibi araçlar kullanılabilir
- Varsayılan durumda web üzerinden sadece **/health** ve **/info** servisleri erişilebilir
- Diğer servislere de **web üzerinden erişim** izni verilebilir

src/main/resources/application.properties



```
management.endpoints.web.exposure.include=*
```

# Actuator Endpointlerine Web Erişimi

- Varsayılan durumda actuator endpoint'lerine erişmek için URI prefix'i olarak **/actuator** eklemek gerekir
- Bu **prefix** değiştirilebilir

src/main/resources/application.properties



```
management.endpoints.web.base-path=/actuator
```

# Bazı Predefined Actuator Endpoint'leri

|                 |   |
|-----------------|---|
| /auditevents    | Login, logout gibi security ile ilgili audit eventleri listeler |
| /beans          | ApplicationContext'de tanımlı bütün bean instance'ları listeler |
| /caches         | Mevcut cache instance'larını listeler                           |
| /configprops    | @ConfigurationProperties bean instance'larını listeler          |
| /env            | Mevcut environment property'lerini listeler                     |
| /health         | Uygulamanın durum bilgisini gösterir                            |
| /heapdump       | JVM heap dump alma imkanı sunar                                 |
| /httptrace      | Son 100 HTTP request-response'a ait trace bilgisini sunar       |
| /info           | Uygulama ile ilgili build, versiyon vb genel bilgi verir        |
| /logfile        | Uygulamada tanımlı log dosyalarına erişim sunar                 |
| /loggers        | Uygulamada tanımlı logger instance'larına erişim sağlar         |
| /mappings       | Tanımlı @RequestMapping'leri listeler                           |
| /metrics        | Uygulama ile ilgili metriklere erişim sağlar                    |
| /scheduledtasks | Uygulamada tanımlı scheduled taskları listeler                  |
| /sessions       | HTTP oturumlarını listeler                                      |
| /shutdown       | Uygulamanın kapatılmasını sağlar                                |
| /threaddump     | JVM thread dump alma imkanı sunar                               |

# Actuator Endpointlerinin Aktivasyonu

- Varsayılan durumda shutdown hariç **bütün endpoint'ler aktiftir**
- İstenirse tek tek, istenirse de topluca bütün hizmetleri **devre dışı** bırakmak mümkündür

src/main/resources/application.properties



```
management.endpoint.health.enabled=false  
management.endpoints.enabled-by-default=false
```

# Mevcut Actuator'ların Tespit Edilmesi

- Endpoint belirtmeden sadece **/actuator** URI'ına erişilirse HTTP üzerinden erişilebilir **bütün actuator endpoint'leri listelenecektir**



# /health Endpoint

- Uygulamanın statü, disk, JMS, DB sunucu durumları gibi bilgilerini sunan endpoint'tir
- Normalde sadece **status:up** şeklinde sınırlı bilgi sunar
- Daha **detaylı bilgi** vermesi de sağlanabilir

application.properties



management.endpoint.health.show-details=**always**

Alabileceği değerler:

- always
- never
- when-authorized

# /health Endpoint

- Health bilgileri **HealthIndicator** arayüzünü implement eden bean'lardan toplanır
- Default olarak register edilmiş HealthIndicator'ları **devre dışı** bırakılabilir

application.properties



```
management.health.defaults.enabled=false
management.health.db.enabled=false
```

# /info Endpoint

- Uygulama ile ilgili build, versiyon vb bilgi sunan endpoint'tir
- Bu bilgiler **InfoContributor** arayüzünü implement eden bean'lardan elde edilir
- **Built-in** InfoContributor'lar mevcuttur, istenirse bunlar **devre dışı** bırakılabilir

application.properties



```
management.info.defaults.enabled=false
management.info.env.enabled=false
```

# /info Endpoint

- Default olarak tanımlı built-in InfoContributor bean'ları şunlardır:
  - **EnvironmentInfoContributor**: info.\* ile başlayan env property'lerine bakar
  - **GitInfoContributor**: classpath:/git.properties dosyasına bakar
  - **BuildInfoContributor**: classpath:/META-INF/build-info.properties dosyasına bakar

# /info Endpoint

- Maven ile çalışırken build-info.properties dosyasının oluşturulabilmesi için pom.xml içerisinde aşağıdaki plugin tanımına ihtiyaç vardır

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>build-info</id>
      <goals>
        <goal>build-info</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

# Endpoints ve CORS Desteği

- Default durumda **CORS desteği devre dışıdır**
- İstenirse **belirtilen domain ve HTTP metotları için** devreye alınabilir

application.properties



```
management.endpoints.web.cors.allowed-origins=http://example.com  
management.endpoints.web.cors.allowed-methods=GET,POST
```

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

