



Spring Boot Actuator

Pertemuan 13

Aplikasi = Kado



- Aplikasi itu ibarat kado, kita bisa melakukan apapun dengan itu dan membuat tebakan apa yang terjadi di dalamnya
- Masalahnya, apa benar tebakan tersebut?
- Semua proses yang ada di dalam aplikasi yang kita jalankan itu tidak terlihat

Lanjutan



- Di pertemuan ini, kita akan membahas Spring Boot Actuator.
- Actuator menyediakan fitur siap-produksi seperti
 - Monitoring
 - Pengukuran Metrik ke aplikasi Spring Boot
- Fitur Actuator disediakan oleh end point sehingga tersedia melalui HTTP maupun JMX MBeans

Actuator



- Di dalam mesin, sebuah actuator adalah komponen yang bertanggung jawab mengendalikan dan menggerakkan mekanisme.
- Di dalam aplikasi Spring Boot pun sama, mengizinkan kita untuk melihat isi dari aplikasi yang dijalankan, dan dalam tingkat tertentu mengendalikan bagaimana aplikasi bertindak

Lanjutan



- Menggunakan endpoint yang dibuka oleh Actuator, kita bisa menanyakan keadaan internal dari aplikasi Spring Boot:
 - Konfigurasi Properti apa yang tersedia di lingkungan aplikasi?
 - Tingkat Logging dari tiap-tiap package di dalam aplikasi?
 - Berapa banyak memory yang digunakan oleh aplikasi?
 - Berapa kali endpoint HTTP telah diminta?
 - Bagaimana kondisi kesehatan dari aplikasi, dan service eksternal nya?

Path Actuator

HTTP method	Path	Description	Enabled by default?
GET	/auditevents	Produces a report of any audit events that have been fired.	No
GET	/beans	Describes all the beans in the Spring application context.	No
GET	/conditions	Produces a report of autoconfiguration conditions that either passed or failed, leading to the beans created in the application context.	No
GET	/configprops	Describes all configuration properties along with the current values.	No
GET, POST, DELETE	/env	Produces a report of all property sources and their properties available to the Spring application.	No
GET	/env/{toMatch}	Describes the value of a single environment property.	No
GET	/health	Returns the aggregate health of the application and (possibly) the health of external dependent applications.	Yes

Lanjutan

HTTP method	Path	Description	Enabled by default?
GET	/heapdump	Downloads a heap dump.	No
GET	/httptrace	Produces a trace of the most recent 100 requests.	No
GET	/info	Returns any developer-defined information about the application.	Yes
GET	/loggers	Produces a list of packages in the application along with their configured and effective logging levels.	No
GET, POST	/loggers/{name}	Returns the configured and effective logging level of a given logger. The effective logging level can be set with a POST request.	No
GET	/mappings	Produces a report of all HTTP mappings and their corresponding handler methods.	No
GET	/metrics	Returns a list of all metrics categories.	No
GET	/metrics/{name}	Returns a multidimensional set of values for a given metric.	No
GET	/scheduledtasks	Lists all scheduled tasks.	No
GET	/threaddump	Returns a report of all application threads.	No

Lanjutan



- Kita ketahui bahwa akses /health dan /info dibuka secara default.
- Sebagian besar endpoints Actuator membawa informasi yang sensitif sehingga perlu diamankan
- Oleh karena itu Spring Security dapat digunakan untuk mengunci Actuator, tetapi karena Actuator tidak mengamankan dirinya sendiri, sehingga sebagian akses endpoint dimatikan, dan perlu konfigurasi manual

Lanjutan



- Jika kita ingin membuka semua endpoints, maka lebih mudah untuk memasukkan semuanya dengan menggunakan wildcard, dan secara eksplisit mengeluarkan yang tidak perlu
- Sebagai Contoh, untuk membuka semua Actuator endpoints kecuali untuk /threaddump dan /heapdump, kita dapat menggunakan kata kunci:
 - `management.endpoints.web.exposure.include`
 - `management.endpoints.web.exposure.exclude`

Contoh



```
management:  
  endpoints:  
    web:  
      exposure:  
        include: '*'  
        exclude: threaddump, heapdump
```

Lanjutan



- Jika kita ingin membuka semua akses selain /health dan /info, maka sebaiknya untuk melakukan konfigurasi Spring Security untuk membatasi akses ke endpoint
- Sekarang kita lihat bagaimana menggunakan endpoint HTTP yang terbuka oleh Actuator

Menggunakan Actuator



- Actuator bisa memberikan informasi yang menarik dan berguna mengenai aplikasi yang berjalan melalui endpoints HTTP
- HTTP endpoint pada dasarnya bisa dianggap seperti REST API, menggunakan klien HTTP apa saja yang diinginkan termasuk Spring RestTemplate dan WebClient, dari browser berbasis aplikasi Javascript atau menggunakan perintah klien curl

Respon Actuator terhadap CURL

```
$ curl localhost:8081/actuator
{
  "_links": {
    "self": {
      "href": "http://localhost:8081/actuator",
      "templated": false
    },
    "auditevents": {
      "href": "http://localhost:8081/actuator/auditevents",
      "templated": false
    },
    "beans": {
      "href": "http://localhost:8081/actuator/beans",
      "templated": false
    },
    "health": {
      "href": "http://localhost:8081/actuator/health",
      "templated": false
    }
  }
}
```


Mengambil Informasi Penting



- Dalam mengambil informasi-informasi yang penting dari sebuah aplikasi, Actuator /info dan /health akan menanyakan selayaknya dokter bertanya kepada pasiennnya
- /info endpoint memberitahu sedikit mengenai aplikasi yang berjalan
- /health endpoint memberitahu sebagaimana sehatnya aplikasi yang berjalan

Curl - /info



- Untuk mengetahui sedikit informasi mengenai aplikasi Spring Boot yang berjalan, kita dapat menanyakannya langsung ke /info endpoint
- Secara default, /info endpoint tidak informatif
- Berikut ini adalah hal kita bisa lihat ketika melakukan request menggunakan curl:

```
$ curl localhost:8081/actuator/info
```

```
{
```

Lanjutan



- Meskipun /info endpoint tidak terlalu berguna, tapi kita cukup menganggapnya sebagai kanvas bersih yang kita bisa isikan informasi yang kita ingin perlihatkan
- Ada beberapa cara untuk memberikan informasi ke /info endpoint, tetapi cara yang paling mudah adalah dengan membuat satu atau lebih konfigurasi properti dengan awalan info..

Lanjutan



info:

contact:

email:phone:support@tacocloud.com
822-625-6831

- Properti-properti di atas tidak terlalu bermakna, namun ketika kita menggunakan curl kembali maka, nilai tersebut akan dikembalikan

Lanjutan



- Curl akan mengembalikan nilai-nilai sebagai hasil dari konfigurasi informasi yang kita ingin lihat

```
{  
  "contact": {"email": "phone": {  
    "support@tacocloud.com",  
    "822-625-6831"  
  }  
}
```


Inspeksi Kesehatan



- Mengisukan permintaan HTTP GET ke /health endpoint akan menghasilkan respon simpel JSON respon dengan informasi kesehatan aplikasi. Sebagai contoh:

```
$ curl localhost:8080/actuator/health  
{"status":"UP"}
```

Lanjutan



- Ternyata status yang diperlihatkan tersebut adalah status agregasi (kumpulan macam-macam status) dari satu maupun lebih informasi kesehatan
- Indikator kesehatan melaporkan kesehatan dari sistem eksternal aplikasi yang berinteraksi dengan databases, message brokers, dan bahkan Spring Cloud components

Lanjutan



- Berikut ini adalah indikator kesehatan
- UP—Sistem Eksternal UP dan Bisa Dicapai
- DOWN—Sistem Eksternal Down dan Tidak Bisa Dicapai
- UNKNOWN—Status sistem tidak jelas
- OUT_OF_SERVICE—Sistem eksternal Bisa Dicapai tetapi tidak tersedia

Lanjutan



- Jika informasi kesehatan teragregasi menjadi satu indikator kesehatan secara penuh, maka berlaku:
 - Jika indikator kesehatan UP, maka kesehatan aplikasi adalah UP.
 - Jika salah satu DOWN, maka kesehatan aplikasi juga DOWN
 - Jika salah satu indikator OUT_OF_SERVICE, maka kesehatan aplikasi juga OUT_OF_SERVICE.
 - Status UNKNOWN diabaikan dan tidak dimasukkan ke dalam agregasi aplikasi

Konfigurasi Actuator

- Salah satu kelebihan terbaik dari Actuator adalah modifikasi yang disesuaikan dengan spesifikasi aplikasi
- Beberapa endpoint bisa dimodifikasi
- Sedangkan, Actuator itu sendiri mengizinkan membuat endpoints modif

Membuat Info Contributor



- Seandainya kita ingin membuat statitika simpel mengenai aplikasi di /info endpoint.
- Sebagai contoh: kita ingin tahu seberapa banyak database sudah diakses
- Untuk melakukannya, kita dapat menulis sebuah kelas yang mengimplementasikan info Contributor yang berguna mempublikasikan informasi tersebut ke /info endpoint

Definisi /health Modif



- Spring Boot datang dengan beberapa indikator kesehatan out-of-the-box yang menyediakan informasi umum sistem eksternal yang aplikasi Spring integrasikan.
- Tetapi dititik tertentu, kita dapat menemukan bahwa interaksi dengan sistem eksternal tidak pernah diantisipasi oleh Spring Boot maupun indikator kesehatan

Membuat Endpoint Sendiri



- Pada dasarnya Endpoint dari Actuator tidak jauh berbeda dari @Controller maupun @RestAPI
- Yang membedakannya adalah anotasi @Endpoint
- Sehingga kita dapat membuat sendiri Actuator Endpoint sesuai dengan kebutuhan kita

Mengamankan Actuator



- Sebenarnya Actuator adalah sesuatu yang orang lain tidak boleh tahu
- Dikarenakan Actuator membawa informasi-informasi sensitif mengenai aplikasi yang sedang berjalan
- Meskipun keamanan itu penting, Actuator tidak menyediakan fitur keamanan
- Tetapi kita dapat menggunakan Spring Security sebagai pengaman

Lanjutan



- Dikarenakan semua endpoint Actuator terkumpul di satu tempat umum yaitu /actuator
- Sehingga lebih mudah untuk menerapkan otorisasi terhadap semua Actuator endpoint yang ada
- Sebagai contoh, diperlukan `ROLE_ADMIN` untuk user ketika mengakses suatu Actuator Endpoint