Rockpedia: Surveillance

Nous voulons toujours savoir si notre application fonctionne correctement, une fois lancée. C'est ce que permettent les outils que nous allons voir.

Surveillance avec Spring Actuator, Prometheus et Spring Boot Admin

Spring Actuator et Prometheus

Spring actuator est déjà activé. Cela permet de voir pas mal d'infos relatives à l'application à l'adresse http://localhost:8080/actuator.



J'ai mis en évidence les plus utiles :

• health: permet de savoir si l'application fonctionne (UP) ou pas

status: "UP"

• info: informations générales sur l'application

```
▼ app:
    name: "Rockpedia"
    description: "L'API qui fait du bruit"
    version: "0.0.1-SNAPSHOT"
    java-vendor: "Oracle Corporation"
```

• prometheus : beaucoup d'informations pas très lisibles, surtout utilisé par des applications externes comme graphana ou Spring Boot Admin, que nous allons voir dans un instant.

```
# HELP hikaricp connections timeout total counter
hikaricp connections timeout total counter
hikaricp connections min some timeout total counter
hikaricp connections min some min some content of the process of the pr
```

metrics: données de surveillance, accessibles à l'adresse
 <u>http://localhost:8080/actuator/metrics/\${DATA}</u> (avec \${DATA} le nom de ce que vous voulez inspecter)

```
▼ names:
    0:
           "jvm.memory.max"
    1:
           "jvm.threads.states"
    2:
           "jdbc.connections.active"
    3:
           "process.files.max"
           "jvm.gc.memory.promoted"
    5:
           "system.load.average.lm"
    6:
           "jvm.memory.used"
    7:
          "jvm.gc.max.data.size"
           "jdbc.connections.max"
    9:
           "jdbc.connections.min"
    10:
           "jvm.gc.pause"
    11:
          "jvm.memory.committed"
    12:
           "system.cpu.count"
    13:
           "logback.events"
    14:
           "http.server.requests"
    15:
           "tomcat.global.sent"
    16:
           "jvm.buffer.memory.used"
    17:
           "tomcat.sessions.created"
    18:
           "jvm.threads.daemon"
    19:
           "system.cpu.usage"
    20:
           "jvm.gc.memory.allocated"
    21:
           "tomcat.global.request.max"
    22:
           "hikaricp.connections.idle"
    23:
           "hikaricp.connections.pending"
    24:
           "tomcat.global.request"
    25:
           "tomcat.sessions.expired"
    26:
           "hikaricp.connections"
    27:
           "jvm.threads.live"
    28:
           "jvm.threads.peak"
    29:
           "tomcat.global.received"
           "hikaricp.connections.active"
    31:
           "hikaricp.connections.creation"
    32:
           "process.uptime"
    33:
           "tomcat.sessions.rejected"
    34:
          "process.cpu.usage"
    35:
           "tomcat.threads.config.max"
           "jvm.classes.loaded"
    36:
    37:
          "hikaricp.connections.max"
    38:
           "hikaricp.connections.min"
    39:
           "jvm.classes.unloaded"
          "tomcat.global.error"
    40:
    41:
           "tomcat.sessions.active.current"
```

Spring Boot Admin

• Clonez le dépot :

git clone https://github.com/SmileEdge/SpringBootAdmin.git

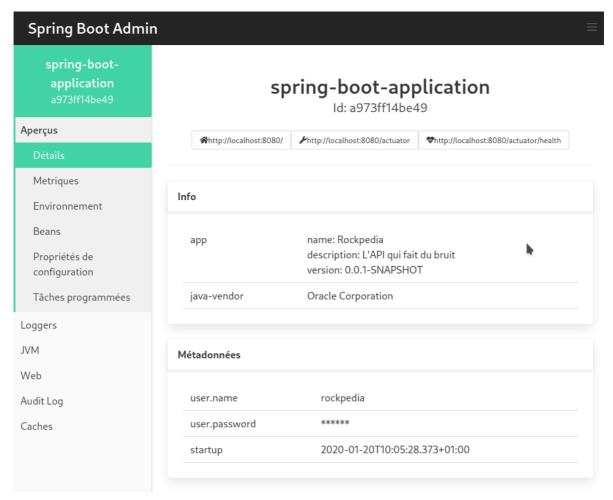
• Build avec maven

mvn build

• Démarrez l'application

mvn spring-boot:run

Rendez vous à l'adresse http://localhost:8081/ avec un navigateur, connectez-vous avec le nom d'utilisateur admin et le mot de passe admin et selectionnez l'application spring-bootapplication en cliquant une fois dessus, puis sur le numéro sous l'adresse http://localhost:8080/.



Sur la page d'accueil, on peut voir les mêmes infos que par la route info de Spring Actuator, le nombre de threads de l'application, la mémoire utilisée, le nombre de passages du garbage collector...

Aperçus > Metriques permet d'ajouter des métriques visibles sur actuator/metrics.

Aperçus > Envirronement liste les variables d'environnement de l'application.

Aperçus > Propriétés de configuration liste d'autres variables d'environement, plus spécifiques à Spring.

Loggers permet de définir la priorité des logs à afficher (DEBUG < FATAL).