Spring Boot 运维管理平台（eladmin）

代码路径：
https://github.com/elunez/eladmin
在线文档
https://eladmin.vip/pages/010101/

新增Actuator 健康检查：
1）添加 依赖 在pom.xml文件中

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

2）application-dev.yaml增加Actuator配置

```
+management:
+  # 可以指定暴露哪些actuator服务，'*'为全部，注意加上引号，被注释的写法表示只允许health，info, metrics, shutdown
+  endpoints:
+    web:
+      exposure:
+        include: health,info, metrics, shutdown
+        # include: '*'  # all
+    # default: http://localhost:8000/actuator/*
+    # base-path: http://localhost:8000/${base-path}/*
+    # base-path: /check (web.base-path)
+  endpoint:
+    # 通过/actuator/shutdown停止服务
+    shutdown:
+      enabled: true
+    # 显示health的详细内容
+    health:
+      show-details: always
+  info: # 显示任意的应用信息，默认关闭，如果是更低一些的版本默认是开启的
+    env:
+      enabled: true
+# 自定义/actuator/info中的各种内容，可以自定义，也可以取默认的一些系统/服务环境变量
+info:
+  app:
+    encoding: "@project.build.sourceEncoding@"
+    java:
+      source: "@java.version@"
+      target: "@java.version@"
+  build:
+    artifact: @project.artifactId@
+    name: @project.name@
+    description: @project.description@
+    pomVersion: @project.version@
+    # 甚至可以自定义test
+    test: 'I love Spring Boot'
```

3）在eladmin-system/src/main/java/me/zhengjie/config/ConfigurerAdapter.java添加注册endpoint逻辑

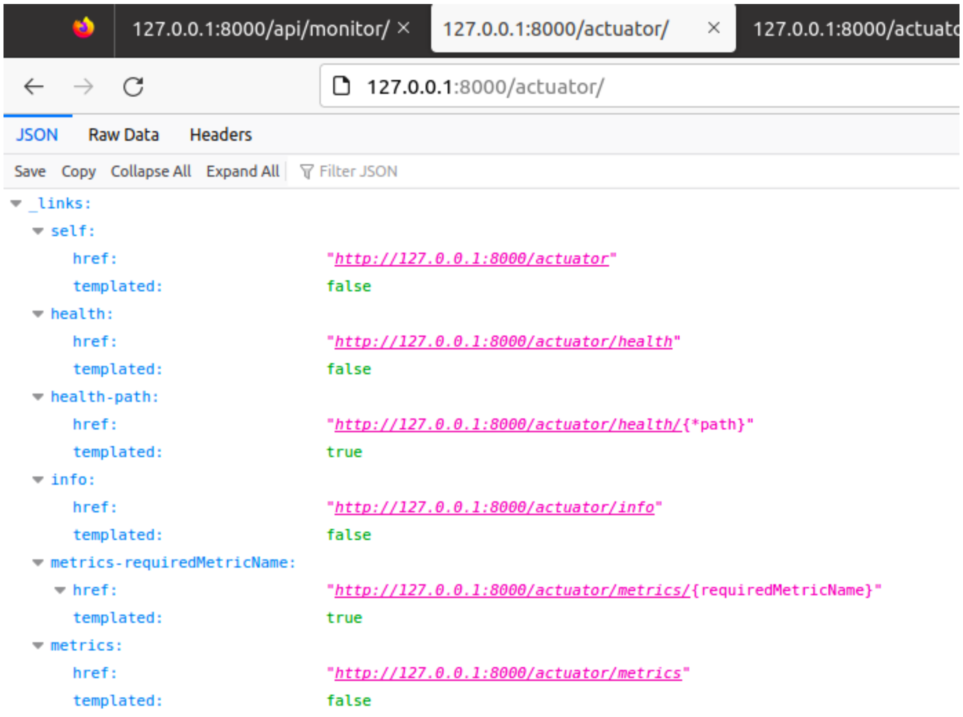
```
+    /**
+    * 注册endpoints,解决springboot升级到2.6.x之后，actuator error problems
+    *
+    * @param webEndpointsSupplier        the web endpoints supplier
+    * @param servletEndpointsSupplier    the servlet endpoints supplier
+    * @param controllerEndpointsSupplier the controller endpoints supplier
+    * @param endpointMediaTypes          the endpoint media types
+    * @param corsProperties              the cors properties
+    * @param webEndpointProperties       the web endpoint properties
+    * @param environment                 the environment
+    * @return the web mvc endpoint handler mapping
+    */
+    @Bean
+    public WebMvcEndpointHandlerMapping webEndpointServletHandlerMapping(WebEndpointsSupplier
webEndpointsSupplier, ServletEndpointsSupplier servletEndpointsSupplier,
+    ControllerEndpointsSupplier controllerEndpointsSupplier, EndpointMediaTypes endpointMediaTypes,
+    CorsEndpointProperties corsProperties, WebEndpointProperties webEndpointProperties,
+    Environment environment) {
+        List<ExposableEndpoint<?>> allEndpoints = new ArrayList<>();
+        Collection<ExposableWebEndpoint> webEndpoints = webEndpointsSupplier.getEndpoints();
+        allEndpoints.addAll(webEndpoints);
+        allEndpoints.addAll(servletEndpointsSupplier.getEndpoints());
+        allEndpoints.addAll(controllerEndpointsSupplier.getEndpoints());
+        String basePath = webEndpointProperties.getBasePath();
+        EndpointMapping endpointMapping = new EndpointMapping(basePath);
+        boolean shouldRegisterLinksMapping = shouldRegisterLinksMapping(webEndpointProperties, environment,
basePath);
+
+    return new WebMvcEndpointHandlerMapping(endpointMapping, webEndpoints, endpointMediaTypes,
+      corsProperties.toCorsConfiguration(), new EndpointLinksResolver(allEndpoints, basePath),
+      shouldRegisterLinksMapping);
+    }
+
+    /**
+    * shouldRegisterLinksMapping
+    * @param webEndpointProperties webEndpointProperties
+    * @param environment environment
+    * @param basePath /
+    * @return boolean
+    */
+    private boolean shouldRegisterLinksMapping(WebEndpointProperties webEndpointProperties,
+    Environment environment, String basePath) {
+    return webEndpointProperties.getDiscovery().isEnabled() && (StringUtils.hasText(basePath)
+      || ManagementPortType.get(environment).equals(ManagementPortType.DIFFERENT));
+    }
```

4)在SpringSecurityConfig.java开放访问权限限制

```
+                .requestMatchers(EndpointRequest.toAnyEndpoint()).permitAll()
```

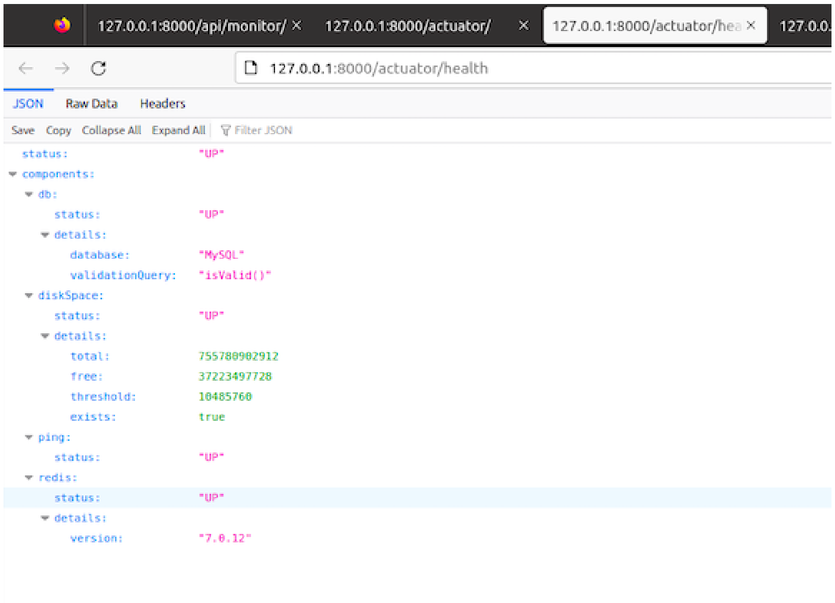
5)效果截图

http://127.0.0.1:8000/actuator



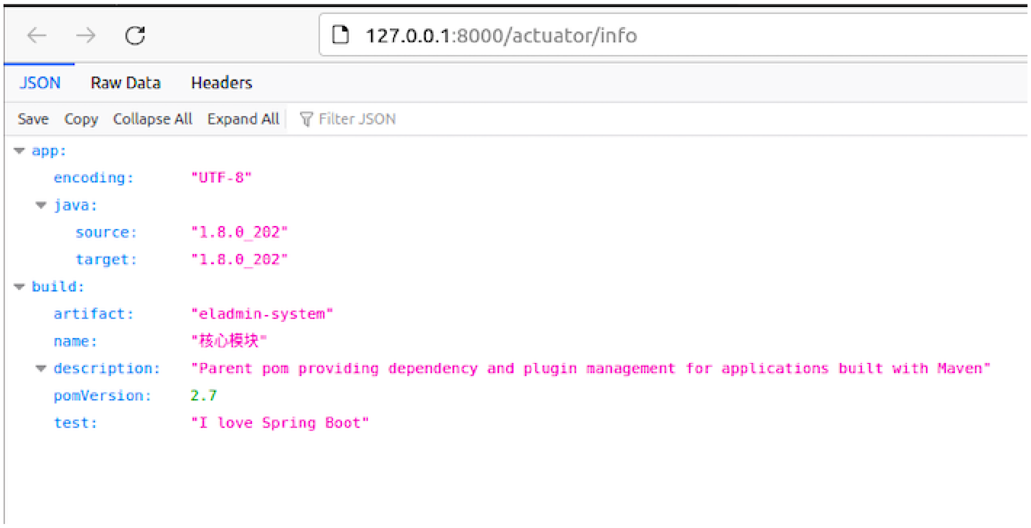
```
▼ _links:
  ▼ self:
      href:                    "http://127.0.0.1:8000/actuator"
      templated:               false
  ▼ health:
      href:                    "http://127.0.0.1:8000/actuator/health"
      templated:               false
  ▼ health-path:
      href:                    "http://127.0.0.1:8000/actuator/health/{*path}"
      templated:               true
  ▼ info:
      href:                    "http://127.0.0.1:8000/actuator/info"
      templated:               false
  ▼ metrics-requiredMetricName:
    ▼ href:                    "http://127.0.0.1:8000/actuator/metrics/{requiredMetricName}"
      templated:               true
  ▼ metrics:
      href:                    "http://127.0.0.1:8000/actuator/metrics"
      templated:               false
```

http://127.0.0.1:8000/actuator/health



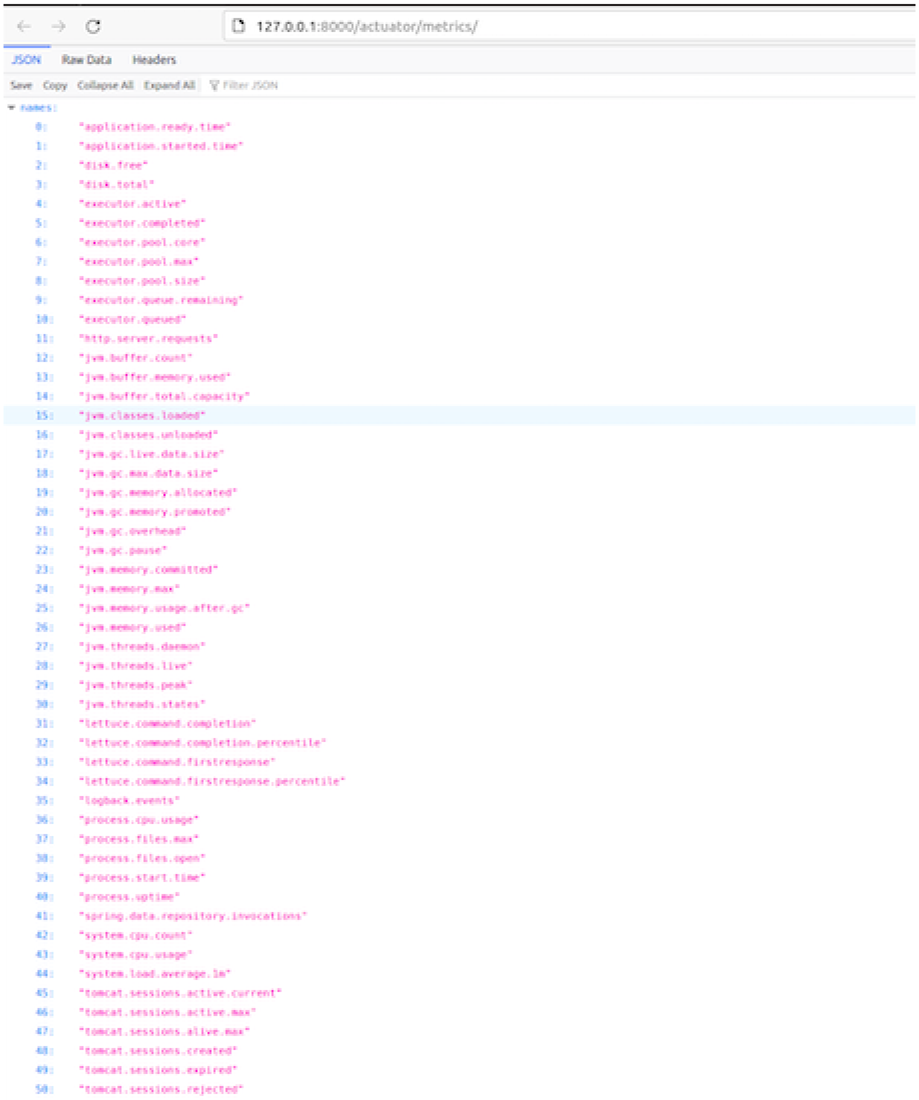
```
  status:                "UP"
▼ components:
  ▼ db:
      status:            "UP"
    ▼ details:
        database:        "MySQL"
        validationQuery: "isValid()"
  ▼ diskSpace:
      status:            "UP"
    ▼ details:
        total:           755780902912
        free:            37223497728
        threshold:       10485760
        exists:          true
  ▼ ping:
      status:            "UP"
  ▼ redis:
      status:            "UP"
    ▼ details:
        version:         "7.0.12"
```

自定义信息 http://127.0.0.1:8000/actuator/info



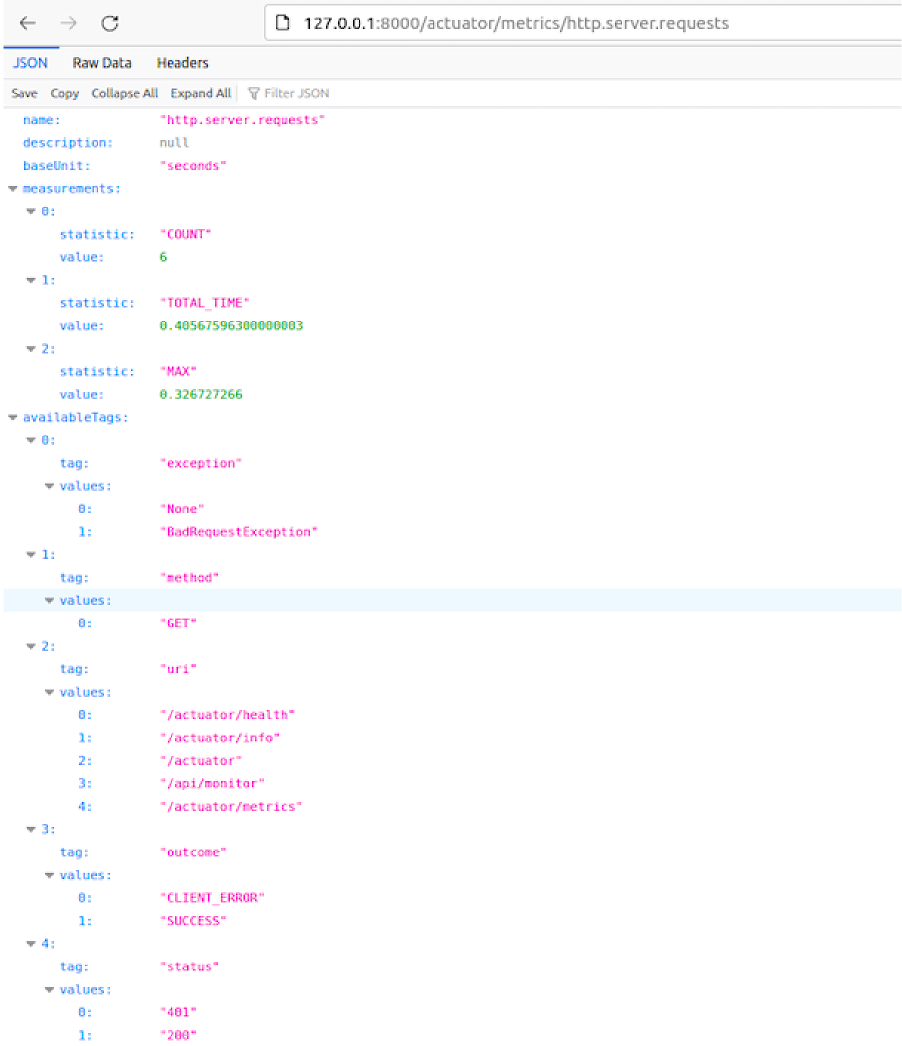
```
▼ app:
    encoding:        "UTF-8"
  ▼ java:
      source:        "1.8.0_202"
      target:        "1.8.0_202"
▼ build:
    artifact:        "eladmin-system"
    name:            "核心模块"
  ▼ description:     "Parent pom providing dependency and plugin management for applications built with Maven"
    pomVersion:      2.7
    test:            "I love Spring Boot"
```

监控项 http://127.0.0.1:8000/actuator/metric



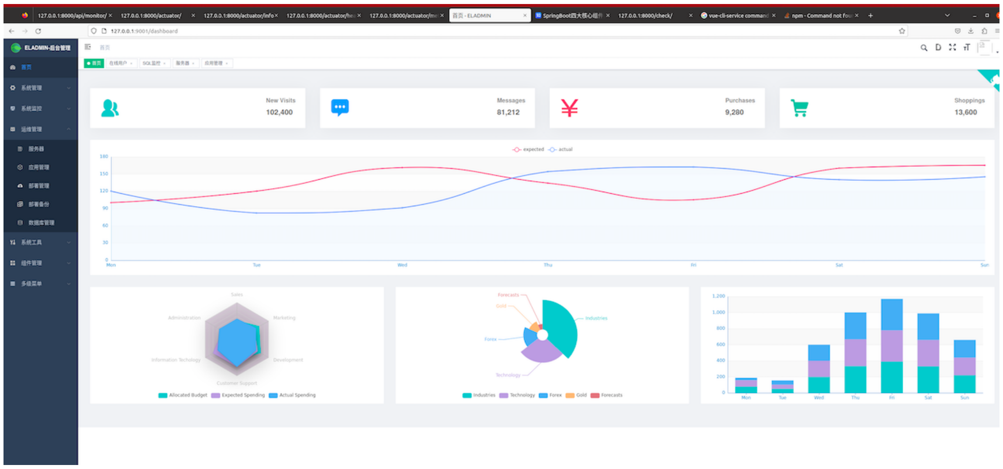
```
▾ names:
    0:   "application.ready.time"
    1:   "application.started.time"
    2:   "disk.free"
    3:   "disk.total"
    4:   "executor.active"
    5:   "executor.completed"
    6:   "executor.pool.core"
    7:   "executor.pool.max"
    8:   "executor.pool.size"
    9:   "executor.queue.remaining"
    10:  "executor.queued"
    11:  "http.server.requests"
    12:  "jvm.buffer.count"
    13:  "jvm.buffer.memory.used"
    14:  "jvm.buffer.total.capacity"
    15:  "jvm.classes.loaded"
    16:  "jvm.classes.unloaded"
    17:  "jvm.gc.live.data.size"
    18:  "jvm.gc.max.data.size"
    19:  "jvm.gc.memory.allocated"
    20:  "jvm.gc.memory.promoted"
    21:  "jvm.gc.overhead"
    22:  "jvm.gc.pause"
    23:  "jvm.memory.committed"
    24:  "jvm.memory.max"
    25:  "jvm.memory.usage.after.gc"
    26:  "jvm.memory.used"
    27:  "jvm.threads.daemon"
    28:  "jvm.threads.live"
    29:  "jvm.threads.peak"
    30:  "jvm.threads.states"
    31:  "lettuce.command.completion"
    32:  "lettuce.command.completion.percentile"
    33:  "lettuce.command.firstresponse"
    34:  "lettuce.command.firstresponse.percentile"
    35:  "logback.events"
    36:  "process.cpu.usage"
    37:  "process.files.max"
    38:  "process.files.open"
    39:  "process.start.time"
    40:  "process.uptime"
    41:  "spring.data.repository.invocations"
    42:  "system.cpu.count"
    43:  "system.cpu.usage"
    44:  "system.load.average.1m"
    45:  "tomcat.sessions.active.current"
    46:  "tomcat.sessions.active.max"
    47:  "tomcat.sessions.alive.max"
    48:  "tomcat.sessions.created"
    49:  "tomcat.sessions.expired"
    50:  "tomcat.sessions.rejected"
```

http://127.0.0.1:8000/actuator/metrics/http.server.request



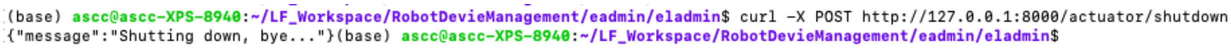
```
name:                "http.server.requests"
description:         null
baseUnit:            "seconds"
▼ measurements:
  ▼ 0:
      statistic:     "COUNT"
      value:         6
  ▼ 1:
      statistic:     "TOTAL_TIME"
      value:         0.405675963000000003
  ▼ 2:
      statistic:     "MAX"
      value:         0.326727266
▼ availableTags:
  ▼ 0:
      tag:           "exception"
    ▼ values:
        0:           "None"
        1:           "BadRequestException"
  ▼ 1:
      tag:           "method"
    ▼ values:
        0:           "GET"
  ▼ 2:
      tag:           "uri"
    ▼ values:
        0:           "/actuator/health"
        1:           "/actuator/info"
        2:           "/actuator"
        3:           "/api/monitor"
        4:           "/actuator/metrics"
  ▼ 3:
      tag:           "outcome"
    ▼ values:
        0:           "CLIENT_ERROR"
        1:           "SUCCESS"
  ▼ 4:
      tag:           "status"
    ▼ values:
        0:           "401"
        1:           "200"
```

Web页面不受影响



6）通过Actuator通过 URL关闭服务

curl -X POST http://127.0.0.1:8000/actuator/shutdown



```
(base) ascc@ascc-XPS-8940:~/LF_Workspace/RobotDevieManagement/eadmin/eladmin$ curl -X POST http://127.0.0.1:8000/actuator/shutdown
{"message":"Shutting down, bye..."}(base) ascc@ascc-XPS-8940:~/LF_Workspace/RobotDevieManagement/eadmin/eladmin$
```

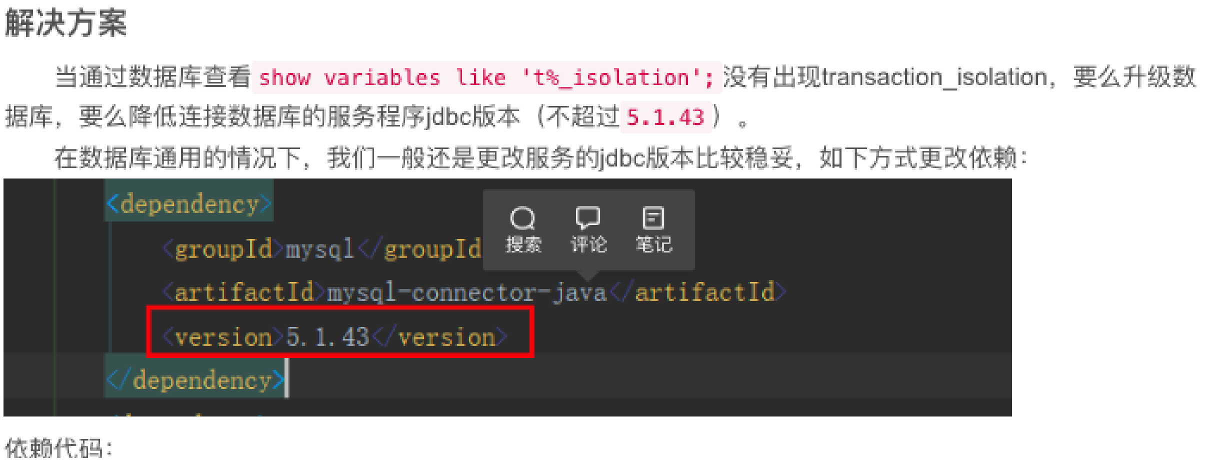
可能遇见的问题：

1）需要init数据库，eladmin表

- 连接数据库(docker 环境）
  docker exec -it mariadb mariadb --user root -p123456
- 创建表
  CREATE DATABASE `eladmin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

2）Caused by: java.sql.SQLException: Unknown system variable ‘transaction_isolation‘

https://blog.csdn.net/Andya_net/article/details/107178920



解决方案

当通过数据库查看`show variables like 't%_isolation';`没有出现transaction_isolation，要么升级数据库，要么降低连接数据库的服务程序jdbc版本（不超过`5.1.43`）。

在数据库通用的情况下，我们一般还是更改服务的jdbc版本比较稳妥，如下方式更改依赖：

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.43</version>
</dependency>
```

依赖代码：

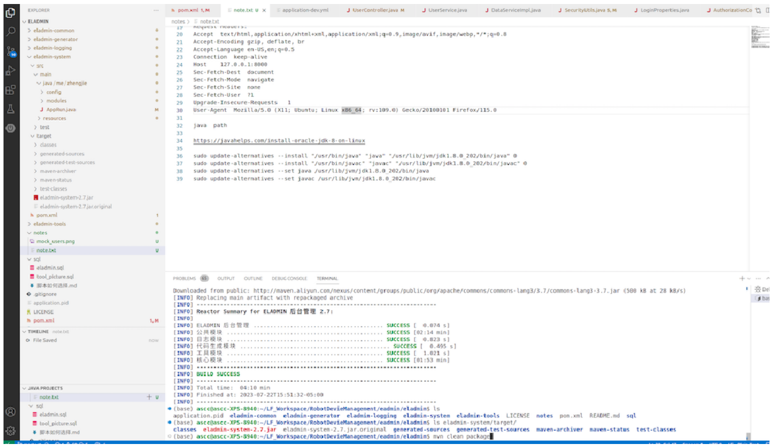
3）Url attribute is not specified and no embedded datasource could be configured

```
<dependency>
   <groupId>com.h2database</groupId>
   <artifactId>h2</artifactId>
</dependency>
```

4）使用Maven打包

Java sdk下载及路径配置(Linux):https://javahelps.com/install-oracle-jdk-8-on-linux

后端服务打包：mvn clean package



```
Request Headers:
Accept  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding gzip, deflate, br
Accept-Language en-US,en;q=0.5
Connection  keep-alive
Host    127.0.0.1:8000
Sec-Fetch-Dest  document
Sec-Fetch-Mode  navigate
Sec-Fetch-Site  none
Sec-Fetch-User  ?1
Upgrade-Insecure-Requests   1
User-Agent  Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

java  path

https://javahelps.com/install-oracle-jdk-8-on-linux

sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_202/bin/java" 0
sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0_202/bin/javac" 0
sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_202/bin/java
sudo update-alternatives --set javac /usr/lib/jvm/jdk1.8.0_202/bin/javac
```

```
Downloaded from public: http://maven.aliyun.com/nexus/content/groups/public/org/apache/commons/commons-lang3/3.7/commons-lang3-3.7.jar (500 kB at 28 kB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for ELADMIN 后台管理 2.7:
[INFO]
[INFO] ELADMIN 后台管理 ................................... SUCCESS [  0.074 s]
[INFO] 公共模块 ........................................... SUCCESS [02:14 min]
[INFO] 日志模块 ........................................... SUCCESS [  0.823 s]
[INFO] 代码生成模块 ......................................... SUCCESS [  0.495 s]
[INFO] 工具模块 ........................................... SUCCESS [  1.021 s]
[INFO] 核心模块 ........................................... SUCCESS [01:53 min]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:10 min
[INFO] Finished at: 2023-07-22T15:51:32-05:00
[INFO] ------------------------------------------------------------------------
(base) ascc@ascc-XPS-8940:~/LF_Workspace/RobotDevieManagement/eadmin/eladmin$ ls
application.pid  eladmin-common  eladmin-generator  eladmin-logging  eladmin-system  eladmin-tools  LICENSE  notes  pom.xml  README.md  sql
(base) ascc@ascc-XPS-8940:~/LF_Workspace/RobotDevieManagement/eadmin/eladmin$ ls eladmin-system/target/
classes  eladmin-system-2.7.jar  eladmin-system-2.7.jar.original  generated-sources  generated-test-sources  maven-archiver  maven-status  test-classes
(base) ascc@ascc-XPS-8940:~/LF_Workspace/RobotDevieManagement/eadmin/eladmin$ mvn clean package
```

前端服务打包:

安装 Node 1.6

```
sudo apt update
curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash
sudo apt-get install -y nodejs
```

打包：npm run build:prod

Reference:

Actuator（英文官网）:
https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#actuator

Actuator参数详解：
https://www.cnblogs.com/caoweixiong/p/15325382.html
https://zhuanlan.zhihu.com/p/603137865

设置Endpoint:
https://zhouj000.github.io/2020/02/19/springboot-6/

SpringBoot2.6,在有spring-boot-starter-actuator依赖的情况下报错，2.5.7正常启动
https://gitee.com/xiaoym/knife4j/issues/I4JT89?skip_mobile=true