Spring Boot Actuator Web API Documentation

Andy Wilkinson

Table of Contents

该 API 文档描述了 Spring Boot Actuators Web 端点.

Chapter 1. 概述

在继续之前，您应该阅读以下主题：

- URLs
- Timestamps

> 为了获得正确的 JSON 响应，下面的 Jackson 必须可用。

1.1. URLs

默认情况下，所有 Web 端点在路径 `/actuator` 下都可用，其URL格式为 `/actuator/{id}`。可以使用 `management.endpoints.web.base-path` 属性配置 `/actuator` 基本路径，如以下示例所示：

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

前面的 `application.properties` 示例将端点URL的形式从 `/actuator/{id}` 更改为 `/manage/{id}`。例如，URL信息端点将变为 `/manage/info`。

1.2. Timestamps

端点消耗的所有时间戳(作为查询参数或在请求正文中)必须格式化为 ISO 8601 中指定的偏移日期和时间。

Chapter 2. 审计事件 (auditevents)

auditevents 端点提供应用程序有关审计事件的信息

2.1. 检索审计事件

要检索审核事件, 请对 /actuator/auditevents 端点发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/auditevents?principal=alice&after=2021-
05-29T15%3A00%3A11.157%2B08%3A00&type=logout' -i -X GET
```

前面的示例检索的是 alice 的 logout 事件, 该事件发生于 2017年11月7日UTC时区.
产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 121

{
  "events" : [ {
    "timestamp" : "2021-05-29T07:00:11.158Z",
    "principal" : "alice",
    "type" : "logout"
  } ]
}
```

2.1.1. 查询参数

端点使用查询参数来限制其返回的事件. 下表显示了支持的查询参数:

Parameter	Description
after	Restricts the events to those that occurred after the given time. Optional.
principal	Restricts the events to those with the given principal. Optional.

Parameter	Description
type	Restricts the events to those with the given type. Optional.

2.1.2. 响应结构

该响应包含与查询匹配的所有审核事件的详细信息. 下表描述了响应的结构:

Path	Type	Description
events	Array	An array of audit events.
events.[].timestamp	String	The timestamp of when the event occurred.
events.[].principal	String	The principal that triggered the event.
events.[].type	String	The type of the event.

Chapter 3. Beans (beans)

beans 端点提供了有关应用程序 beans 的详细信息

3.1. 检索 Beans

要检索 beans，请向 /actuator/beans 端点发送 GET 请求，如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/beans' -i -X GET
```

响应结果如下所示

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 1089

{
  "contexts" : {
    "application" : {
      "beans" : {

"org.springframework.boot.autoconfigure.web.servlet.DispatcherServletAutoConfigu
ration$DispatcherServletRegistrationConfiguration" : {
          "aliases" : [ ],
          "scope" : "singleton",
          "type" :
"org.springframework.boot.autoconfigure.web.servlet.DispatcherServletAutoConfigu
ration$DispatcherServletRegistrationConfiguration",
          "dependencies" : [ ]
        },

"org.springframework.boot.autoconfigure.context.PropertyPlaceholderAutoConfigura
tion" : {
          "aliases" : [ ],
          "scope" : "singleton",
          "type" :
"org.springframework.boot.autoconfigure.context.PropertyPlaceholderAutoConfigura
tion",
          "dependencies" : [ ]
        },

"org.springframework.boot.autoconfigure.web.servlet.DispatcherServletAutoConfigu
ration" : {
          "aliases" : [ ],
          "scope" : "singleton",
          "type" :
"org.springframework.boot.autoconfigure.web.servlet.DispatcherServletAutoConfigu
ration",
          "dependencies" : [ ]
        }
      }
    }
  }
}
```

3.1.1. 响应结构

响应包含应用程序bean的详细信息. 下表描述了响应的结构:

Path	Type	Description
`contexts`	`Object`	Application contexts keyed by id.
`contexts.*.parentId`	`String`	Id of the parent application context, if any.
`contexts.*.beans`	`Object`	Beans in the application context keyed by name.
`contexts.*.beans.*.aliases`	`Array`	Names of any aliases.
`contexts.*.beans.*.scope`	`String`	Scope of the bean.
`contexts.*.beans.*.type`	`String`	Fully qualified type of the bean.
`contexts.*.beans.*.resource`	`String`	Resource in which the bean was defined, if any.
`contexts.*.beans.*.dependencies`	`Array`	Names of any dependencies.

Chapter 4. Caches (caches)

caches 端点提供了有关应用程序科访问 caches 的详细信息

4.1. 检索所有的 Caches

要检索所有的 caches, 请向 /actuator/caches 端点发送 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/caches' -i -X GET
```

响应结果如下所示

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 435

{
  "cacheManagers" : {
    "anotherCacheManager" : {
      "caches" : {
        "countries" : {
          "target" : "java.util.concurrent.ConcurrentHashMap"
        }
      }
    },
    "cacheManager" : {
      "caches" : {
        "cities" : {
          "target" : "java.util.concurrent.ConcurrentHashMap"
        },
        "countries" : {
          "target" : "java.util.concurrent.ConcurrentHashMap"
        }
      }
    }
  }
}
```

4.1.1. 响应结构

该响应包含应用程序缓存的详细信息. 下表描述了响应的结构:

Path	Type	Description
`cacheManagers`	`Object`	Cache managers keyed by id.
`cacheManagers.*.caches`	`Object`	Caches in the application context keyed by name.
`cacheManagers.*.caches.*.target`	`String`	Fully qualified name of the native cache.

4.2. 通过 name 检索缓存

要按名称检索缓存, 请向 `/actuator/caches/{name}` 发出 `GET` 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/caches/cities' -i -X GET
```

前面的示例检索有关名为 `cities` 的缓存的信息. 产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 113

{
  "target" : "java.util.concurrent.ConcurrentHashMap",
  "name" : "cities",
  "cacheManager" : "cacheManager"
}
```

4.2.1. 查询参数

如果请求的名称足够代表一个缓存, 则不需要额外的参数. 否则, `cacheManager` 必须指定. 下表显示了受支持的查询参数:

Parameter	Description
cacheManager	Name of the cacheManager to qualify the cache. May be omitted if the cache name is unique.

4.2.2. 响应结构

该响应包含请求的缓存的详细信息. 下表描述了响应的结构:

Path	Type	Description
name	String	Cache name.
cacheManager	String	Cache manager name.
target	String	Fully qualified name of the native cache.

4.3. 删除所有缓存

要清除所有可用的缓存, 请向 /actuator/caches 发送 DELETE 请求, 下面的基于 curl 的示例所示发出请求:

```
$ curl 'http://localhost:8080/actuator/caches' -i -X DELETE
```

4.4. 按名称删除缓存

要删除特定的缓存, 请向 /actuator/caches/{name} 发送 DELETE 请求, 以下基于 curl的示例中所示发出请求:

```
$ curl
'http://localhost:8080/actuator/caches/countries?cacheManager=anotherCacheManage
r' -i -X DELETE
```

由于有两个名为 countries 的缓存, 因此 cacheManager 必须提供来指定 Cache 应清除的缓存.

4.4.1. 请求结构

如果请求的名称足够标识单个缓存，则不需要额外的参数．否则，`cacheManager` 必须指定．下表显示了受支持的查询参数：

Parameter	Description
`cacheManager`	Name of the cacheManager to qualify the cache. May be omitted if the cache name is unique.

Chapter 5. 条件评估报告 (conditions)

conditions 端点提供有关的配置和自动配置类条件的评估信息.

5.1. 检索报告

要检索报告, 请向 /actuator/conditions 发送 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/conditions' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 3255

{
  "contexts" : {
    "application" : {
      "positiveMatches" : {
        "EndpointAutoConfiguration#endpointOperationParameterMapper" : [ {
          "condition" : "OnBeanCondition",
          "message" : "@ConditionalOnMissingBean (types:
org.springframework.boot.actuate.endpoint.invoke.ParameterValueMapper;
SearchStrategy: all) did not find any beans"
        } ],
        "EndpointAutoConfiguration#endpointCachingOperationInvokerAdvisor" : [ {
          "condition" : "OnBeanCondition",
          "message" : "@ConditionalOnMissingBean (types:
org.springframework.boot.actuate.endpoint.invoker.cache.CachingOperationInvokerA
dvisor; SearchStrategy: all) did not find any beans"
        } ],
        "WebEndpointAutoConfiguration" : [ {
          "condition" : "OnWebApplicationCondition",
          "message" : "@ConditionalOnWebApplication (required) found 'session'
scope"
        } ]
      },
      "negativeMatches" : {
        "WebFluxEndpointManagementContextConfiguration" : {
          "notMatched" : [ {
            "condition" : "OnWebApplicationCondition",
            "message" : "not a reactive web application"
          } ],
          "matched" : [ {
            "condition" : "OnClassCondition",
            "message" : "@ConditionalOnClass found required classes
'org.springframework.web.reactive.DispatcherHandler',
'org.springframework.http.server.reactive.HttpHandler'"
          } ]
        },

"GsonHttpMessageConvertersConfiguration.GsonHttpMessageConverterConfiguration" :
{
          "notMatched" : [ {
            "condition" :
"GsonHttpMessageConvertersConfiguration.PreferGsonOrJacksonAndJsonbUnavailableCo
ndition",
            "message" : "AnyNestedCondition 0 matched 2 did not; NestedCondition
on
GsonHttpMessageConvertersConfiguration.PreferGsonOrJacksonAndJsonbUnavailableCon
dition.JacksonJsonbUnavailable NoneNestedConditions 1 matched 1 did not;
NestedCondition on
GsonHttpMessageConvertersConfiguration.JacksonAndJsonbUnavailableCondition.Jsonb
Preferred @ConditionalOnProperty (spring.mvc.converters.preferred-json-
mapper=jsonb) did not find property 'spring.mvc.converters.preferred-json-
mapper'; NestedCondition on
GsonHttpMessageConvertersConfiguration.JacksonAndJsonbUnavailableCondition.Jacks
onAvailable @ConditionalOnBean (types:
org.springframework.http.converter.json.MappingJackson2HttpMessageConverter;
SearchStrategy: all) found bean 'mappingJackson2HttpMessageConverter';
NestedCondition on
GsonHttpMessageConvertersConfiguration.PreferGsonOrJacksonAndJsonbUnavailableCon
dition.GsonPreferred @ConditionalOnProperty (spring.mvc.converters.preferred-
json-mapper=gson) did not find property 'spring.mvc.converters.preferred-json-
mapper'"
          } ],
          "matched" : [ ]
        },
        "JsonbHttpMessageConvertersConfiguration" : {
          "notMatched" : [ {
            "condition" : "OnClassCondition",
            "message" : "@ConditionalOnClass did not find required class
'javax.json.bind.Jsonb'"
          } ],
          "matched" : [ ]
        }
      },
      "unconditionalClasses" : [
"org.springframework.boot.autoconfigure.context.PropertyPlaceholderAutoConfigura
tion",
"org.springframework.boot.actuate.autoconfigure.endpoint.EndpointAutoConfigurati
on" ]
    }
  }
}
```

5.1.1. 响应结构

该响应包含应用程序条件评估的详细信息. 下表描述了响应的结构:

Path	Type	Description
`contexts`	`Object`	Application contexts keyed by id.
`contexts.*.positiveMatches`	`Object`	Classes and methods with conditions that were matched.
`contexts.*.positiveMatches.*.[].condition`	`String`	Name of the condition.
`contexts.*.positiveMatches.*.[].message`	`String`	Details of why the condition was matched.
`contexts.*.negativeMatches`	`Object`	Classes and methods with conditions that were not matched.
`contexts.*.negativeMatches.*.notMatched`	`Array`	Conditions that were matched.
`contexts.*.negativeMatches.*.notMatched.[].condition`	`String`	Name of the condition.
`contexts.*.negativeMatches.*.notMatched.[].message`	`String`	Details of why the condition was not matched.
`contexts.*.negativeMatches.*.matched`	`Array`	Conditions that were matched.
`contexts.*.negativeMatches.*.matched.[].condition`	`String`	Name of the condition.

Path	Type	Description
`contexts.*.negativeMatches.*.matched.[].message`	`String`	Details of why the condition was matched.
`contexts.*.unconditionalClasses`	`Array`	Names of unconditional auto-configuration classes if any.
`contexts.*.parentId`	`String`	Id of the parent application context, if any.

Chapter 6. 配置属性 (configprops)

configprops 端点提供有关应用程序 @ConfigurationProperties beans 的信息.

6.1. 检索 @ConfigurationProperties Bean

要检索 @ConfigurationProperties beans, 请向 /actuator/configprops 发送 GET 请求, 如以下基于curl的示例所示:

```
$ curl 'http://localhost:8080/actuator/configprops' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 3178

{
  "contexts" : {
    "application" : {
      "beans" : {
        "management.endpoints.web.cors-
org.springframework.boot.actuate.autoconfigure.endpoint.web.CorsEndpointProperti
es" : {
          "prefix" : "management.endpoints.web.cors",
          "properties" : {
            "allowedHeaders" : [ ],
            "allowedMethods" : [ ],
            "allowedOrigins" : [ ],
            "maxAge" : "PT30M",
            "exposedHeaders" : [ ]
          },
          "inputs" : {
            "allowedHeaders" : [ ],
            "allowedMethods" : [ ],
            "allowedOrigins" : [ ],
            "maxAge" : { },
            "exposedHeaders" : [ ]
          }
        },
        "management.endpoints.web-
org.springframework.boot.actuate.autoconfigure.endpoint.web.WebEndpointPropertie
```

```
s" : {
          "prefix" : "management.endpoints.web",
          "properties" : {
            "pathMapping" : { },
            "exposure" : {
              "include" : [ "*" ],
              "exclude" : [ ]
            },
            "basePath" : "/actuator"
          },
          "inputs" : {
            "pathMapping" : { },
            "exposure" : {
              "include" : [ {
                "value" : "*",
                "origin" : "\"management.endpoints.web.exposure.include\" from
property source \"Inlined Test Properties\""
              } ],
              "exclude" : [ ]
            },
            "basePath" : { }
          }
        },
        "spring.web-org.springframework.boot.autoconfigure.web.WebProperties" :
{
          "prefix" : "spring.web",
          "properties" : {
            "localeResolver" : "ACCEPT_HEADER",
            "resources" : {
              "staticLocations" : [ "classpath:/META-INF/resources/",
"classpath:/resources/", "classpath:/static/", "classpath:/public/" ],
              "addMappings" : true,
              "chain" : {
                "cache" : true,
                "compressed" : false,
                "strategy" : {
                  "fixed" : {
                    "enabled" : false,
                    "paths" : [ "/**" ]
                  },
                  "content" : {
                    "enabled" : false,
                    "paths" : [ "/**" ]
                  }
                }
              },
              "cache" : {
                "cachecontrol" : { },
                  "useLastModified" : true
                }
              }
            },
            "inputs" : {
              "localeResolver" : { },
              "resources" : {
                "staticLocations" : [ { }, { }, { }, { } ],
                "addMappings" : { },
                "chain" : {
                  "cache" : { },
                  "compressed" : { },
                  "strategy" : {
                    "fixed" : {
                      "enabled" : { },
                      "paths" : [ { } ]
                    },
                    "content" : {
                      "enabled" : { },
                      "paths" : [ { } ]
                    }
                  }
                },
                "cache" : {
                  "cachecontrol" : { },
                  "useLastModified" : { }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

6.1.1. 响应结构

该响应包含应用程序 `@ConfigurationProperties` Bean 的详细信息.
下表描述了响应的结构:

Path	Type	Description
`contexts`	`Object`	Application contexts keyed by id.

Path	Type	Description
`contexts.*.beans.*`	`Object`	`@ConfigurationProperties` beans keyed by bean name.
`contexts.*.beans.*.prefix`	`String`	Prefix applied to the names of the bean's properties.
`contexts.*.beans.*.properties`	`Object`	Properties of the bean as name-value pairs.
`contexts.*.beans.*.inputs`	`Object`	Origin and value of the configuration property used when binding to this bean.
`contexts.*.parentId`	`String`	Id of the parent application context, if any.

Chapter 7. 环境 (env)

env 端点提供有关应用程序 Environment 的信息 .

7.1. 检索整个环境

要检索整个环境, 请向 /actuator/env 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/env' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 836

{
  "activeProfiles" : [ ],
  "propertySources" : [ {
    "name" : "systemProperties",
    "properties" : {
      "java.runtime.name" : {
        "value" : "OpenJDK Runtime Environment"
      },
      "java.vm.version" : {
        "value" : "25.292-b10"
      },
      "java.vm.vendor" : {
        "value" : "AdoptOpenJDK"
      }
    }
  }, {
    "name" : "systemEnvironment",
    "properties" : {
      "JAVA_HOME" : {
        "value" : "/usr/local/jdk8u292-b10",
        "origin" : "System Environment Property \"JAVA_HOME\""
      }
    }
  }, {
    "name" : "Config resource 'class path resource [application.properties]' via
location 'classpath:/'",
    "properties" : {
      "com.example.cache.max-size" : {
        "value" : "1000",
        "origin" : "class path resource [application.properties] - 1:29"
      }
    }
  } ]
}
```

7.1.1. 响应结构

响应包含应用程序 `Environment` 的详细信息 . 下表描述了响应的结构:

Path	Type	Description
activeProfiles	Array	Names of the active profiles, if any.
propertySources	Array	Property sources in order of precedence.
propertySources.[].name	String	Name of the property source.
propertySources.[].properties	Object	Properties in the property source keyed by property name.
propertySources.[].properties.*.value	String	Value of the property.
propertySources.[].properties.*.origin	String	Origin of the property, if any.

7.2. 检索单个属性

要检索单个属性，请向 `/actuator/env/{property.name}` 发出 `GET` 请求，如以下基于 `curl` 的示例所示：

```
$ curl 'http://localhost:8080/actuator/env/com.example.cache.max-size' -i -X GET
```

前面的示例检索有关名为的属性的信息 `com.example.cache.max-size`.
产生的响应类似于以下内容：

```
HTTP/1.1 200 OK
Content-Disposition: inline;filename=f.txt
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 517

{
  "property" : {
    "source" : "Config resource 'class path resource [application.properties]'
via location 'classpath:/'",
    "value" : "1000"
  },
  "activeProfiles" : [ ],
  "propertySources" : [ {
    "name" : "systemProperties"
  }, {
    "name" : "systemEnvironment"
  }, {
    "name" : "Config resource 'class path resource [application.properties]' via
location 'classpath:/'",
    "property" : {
      "value" : "1000",
      "origin" : "class path resource [application.properties] - 1:29"
    }
  } ]
}
```

7.2.1. 响应结构

该响应包含所请求属性的详细信息. 下表描述了响应的结构:

Path	Type	Description
`property`	`Object`	Property from the environment, if found.
`property.source`	`String`	Name of the source of the property.
`property.value`	`String`	Value of the property.
`activeProfiles`	`Array`	Names of the active profiles, if any.

Path	Type	Description
`propertySources`	`Array`	Property sources in order of precedence.
`propertySources.[].name`	`String`	Name of the property source.
`propertySources.[].property`	`Object`	Property in the property source, if any.
`propertySources.[].property.value`	`Varies`	Value of the property.
`propertySources.[].property.origin`	`String`	Origin of the property, if any.

Chapter 8. Flyway (flyway)

flyway 端点提供了有关 Flyway 数据库迁移的信息.

8.1. 检索 Migrations

要检索 migrations, 请向 /actuator/flyway 发出 GET 请求 , 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/flyway' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 515

{
  "contexts" : {
    "application" : {
      "flywayBeans" : {
        "flyway" : {
          "migrations" : [ {
            "type" : "SQL",
            "checksum" : -156244537,
            "version" : "1",
            "description" : "init",
            "script" : "V1__init.sql",
            "state" : "SUCCESS",
            "installedBy" : "SA",
            "installedOn" : "2021-05-29T07:00:15.507Z",
            "installedRank" : 1,
            "executionTime" : 4
          } ]
        }
      }
    }
  }
}
```

8.1.1. 响应结构

该响应包含应用程序的Flyway迁移的详细信息. 下表描述了响应的结构:

Path	Type	Description
contexts	Object	Application contexts keyed by id
contexts.*.flywayBeans.*.migrations	Array	Migrations performed by the Flyway instance, keyed by Flyway bean name.
contexts.*.flywayBeans.*.migrations.[].checksum	Number	Checksum of the migration, if any.
contexts.*.flywayBeans.*.migrations.[].description	String	Description of the migration, if any.
contexts.*.flywayBeans.*.migrations.[].executionTime	Number	Execution time in milliseconds of an applied migration.
contexts.*.flywayBeans.*.migrations.[].installedBy	String	User that installed the applied migration, if any.
contexts.*.flywayBeans.*.migrations.[].installedOn	String	Timestamp of when the applied migration was installed, if any.
contexts.*.flywayBeans.*.migrations.[].installedRank	Number	Rank of the applied migration, if any. Later migrations have higher ranks.
contexts.*.flywayBeans.*.migrations.[].script	String	Name of the script used to execute the migration, if any.

Path	Type	Description
contexts.*.flywayBeans.*.migrations.[].state	String	State of the migration. (PENDING, ABOVE_TARGET, BELOW_BASELINE, BASELINE, IGNORED, MISSING_SUCCESS, MISSING_FAILED, SUCCESS, UNDONE, AVAILABLE, FAILED, OUT_OF_ORDER, FUTURE_SUCCESS, FUTURE_FAILED, OUTDATED, SUPERSEDED, DELETED)
contexts.*.flywayBeans.*.migrations.[].type	String	Type of the migration. (SCHEMA, BASELINE, DELETE, SQL, UNDO_SQL, JDBC, UNDO_JDBC, SPRING_JDBC, UNDO_SPRING_JDBC, CUSTOM, UNDO_CUSTOM)
contexts.*.flywayBeans.*.migrations.[].version	String	Version of the database after applying the migration, if any.
contexts.*.parentId	String	Id of the parent application context, if any.

Chapter 9. Health (health)

health 端点提供有关应用程序的运行状况的详细信息.

9.1. 检索应用程序的运行状况

要检索应用程序的运行状况, 请向 /actuator/health 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/health' -i -X GET \
    -H 'Accept: application/json'
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 702

{
  "status" : "UP",
  "components" : {
    "broker" : {
      "status" : "UP",
      "components" : {
        "us1" : {
          "status" : "UP",
          "details" : {
            "version" : "1.0.2"
          }
        },
        "us2" : {
          "status" : "UP",
          "details" : {
            "version" : "1.0.4"
          }
        }
      }
    },
    "db" : {
      "status" : "UP",
      "details" : {
        "database" : "H2",
        "validationQuery" : "isValid()"
      }
    },
    "diskSpace" : {
      "status" : "UP",
      "details" : {
        "total" : 77397430272,
        "free" : 73527320576,
        "threshold" : 10485760,
        "exists" : true
      }
    }
  }
}
```

9.1.1. 响应结构

该响应包含应用程序运行状况的详细信息. 下表描述了响应的结构:

Path	Type	Description
status	String	Overall status of the application.
components	Object	The components that make up the health.
components.*.status	String	Status of a specific part of the application.
components.*.components	Object	The nested components that make up the health.
components.*.details	Object	Details of the health of a specific part of the application. Presence is controlled by management.endpoint.health.show-details.

上面的响应字段适用于 V3 API. 如果您需要返回 V2 JSON, 则应使用 accept 头或 application/vnd.spring-boot.actuator.v2+json

9.2. 检索组件的运行状况

要检索应用程序运行状况的特定组件的运行状况, 请向 /actuator/health/{component} 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/health/db' -i -X GET \
    -H 'Accept: application/json'
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 101

{
  "status" : "UP",
  "details" : {
    "database" : "H2",
    "validationQuery" : "isValid()"
  }
}
```

9.2.1. 响应结构

该响应包含应用程序中特定组件的运行状况的详细信息. 下表描述了响应的结构:

Path	Type	Description
`status`	`String`	Status of a specific part of the application
`details`	`Object`	Details of the health of a specific part of the application.

9.3. 检索嵌套组件的运行状况

如果特定组件包含其他嵌套组件(如上例中的 `broker` 指标), 则可以通过向 `/actuator/health/{component}/{subcomponent}` 发出 `GET` 请求来检索此类嵌套组件的运行状况, 以下基于 `curl` 的示例所示:

```
$ curl 'http://localhost:8080/actuator/health/broker/us1' -i -X GET \
    -H 'Accept: application/json'
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 66

{
  "status" : "UP",
  "details" : {
    "version" : "1.0.2"
  }
}
```

应用程序运行状况的组件可以任意深度嵌套，
具体取决于应用程序的运行状况指示器及其分组方式。运行状况端点支持 `/{component}` URL中的任意数量的标识符，以允许检索任何深度的组件的运行状况。

9.3.1. 响应结构

该响应包含应用程序特定组件实例的运行状况的详细信息。下表描述了响应的结构：

Path	Type	Description
`status`	`String`	Status of a specific part of the application
`details`	`Object`	Details of the health of a specific part of the application.

Chapter 10. Heap Dump (heapdump)

heapdump 端点从应用程序的 JVM 提供了一个堆转储文件.

10.1. 检索 Heap Dump

要检索堆转储文件, 请向 /actuator/heapdump 发出 GET 请求. 响应是 HPROF 格式的二进制数据, 并且可能很大. 通常, 您应将响应保存到磁盘以进行后续分析. 使用 curl 时, 可以通过使用该 -O 选项来实现, 如以下示例所示:

```
$ curl 'http://localhost:8080/actuator/heapdump' -O
```

前面的示例将一个名为的文件 heapdump 写入当前工作目录.

Chapter 11. HTTP 跟踪 (httptrace)

httptrace 端点提供关于 HTTP 请求-响应交换信息.

11.1. 检索 Traces

要检索跟踪, 请向 /actuator/httptrace 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/httptrace' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 503

{
  "traces" : [ {
    "timestamp" : "2021-05-29T07:00:17.227Z",
    "principal" : {
      "name" : "alice"
    },
    "session" : {
      "id" : "c38b0f53-909e-4e1d-b628-4b69c7dfbe1a"
    },
    "request" : {
      "method" : "GET",
      "uri" : "https://api.example.com",
      "headers" : {
        "Accept" : [ "application/json" ]
      }
    },
    "response" : {
      "status" : 200,
      "headers" : {
        "Content-Type" : [ "application/json" ]
      }
    },
    "timeTaken" : 1
  } ]
}
```

11.1.1. 响应结构

响应包含跟踪的 HTTP 请求-响应交换的详细信息. 下表描述了响应的结构:

Path	Type	Description
`traces`	`Array`	An array of traced HTTP request-response exchanges.
`traces.[].timestamp`	`String`	Timestamp of when the traced exchange occurred.
`traces.[].principal`	`Object`	Principal of the exchange, if any.

Path	Type	Description
`traces.[].principal.name`	`String`	Name of the principal.
`traces.[].request.method`	`String`	HTTP method of the request.
`traces.[].request.remoteAddress`	`String`	Remote address from which the request was received, if known.
`traces.[].request.uri`	`String`	URI of the request.
`traces.[].request.headers`	`Object`	Headers of the request, keyed by header name.
`traces.[].request.headers.*.[]`	`Array`	Values of the header
`traces.[].response.status`	`Number`	Status of the response
`traces.[].response.headers`	`Object`	Headers of the response, keyed by header name.
`traces.[].response.headers.*.[]`	`Array`	Values of the header
`traces.[].session`	`Object`	Session associated with the exchange, if any.
`traces.[].session.id`	`String`	ID of the session.
`traces.[].timeTaken`	`Number`	Time, in milliseconds, taken to handle the exchange.

Chapter 12. Info (info)

info 端点提供有关应用程序的一般信息.

12.1. 检索信息

要检索有关应用程序的信息, 请向 /actuator/info 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/info' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 233

{
  "git" : {
    "commit" : {
      "time" : "+53377-09-27T20:52:19Z",
      "id" : "df027cf"
    },
    "branch" : "main"
  },
  "build" : {
    "version" : "1.0.3",
    "artifact" : "application",
    "group" : "com.example"
  }
}
```

12.1.1. 响应结构

该响应包含有关该应用程序的常规信息. 响应的每个部分均由 InfoContributor 提供. Spring Boot 提供 build 和 git contributions.

build 响应结构

下表描述 `build` 了响应部分的结构:

Path	Type	Description
`artifact`	`String`	Artifact ID of the application, if any.
`group`	`String`	Group ID of the application, if any.
`name`	`String`	Name of the application, if any.
`version`	`String`	Version of the application, if any.
`time`	`Varies`	Timestamp of when the application was built, if any.

Build 响应结构

下表描述 `git` 了响应部分的结构:

Path	Type	Description
`branch`	`String`	Name of the Git branch, if any.
`commit`	`Object`	Details of the Git commit, if any.
`commit.time`	`Varies`	Timestamp of the commit, if any.
`commit.id`	`String`	ID of the commit, if any.

Chapter 13. Spring Integration graph (integrationgraph)

integrationgraph 端点暴露包含所有 Spring Integration graph.

13.1. 检索 Spring Integration graph

要检索有关应用程序的信息，请向 /actuator/integrationgraph 发出 GET 请求，如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/integrationgraph' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 961

{
  "contentDescriptor" : {
    "providerVersion" : "5.4.6",
    "providerFormatVersion" : 1.2,
    "provider" : "spring-integration"
  },
  "nodes" : [ {
    "nodeId" : 1,
    "componentType" : "null-channel",
    "integrationPatternType" : "null_channel",
    "integrationPatternCategory" : "messaging_channel",
    "properties" : { },
    "name" : "nullChannel"
  }, {
    "nodeId" : 2,
    "componentType" : "publish-subscribe-channel",
    "integrationPatternType" : "publish_subscribe_channel",
    "integrationPatternCategory" : "messaging_channel",
    "properties" : { },
    "name" : "errorChannel"
  }, {
    "nodeId" : 3,
    "componentType" : "logging-channel-adapter",
    "integrationPatternType" : "outbound_channel_adapter",
    "integrationPatternCategory" : "messaging_endpoint",
    "properties" : { },
    "input" : "errorChannel",
    "name" : "errorLogger"
  } ],
  "links" : [ {
    "from" : 2,
    "to" : 3,
    "type" : "input"
  } ]
}
```

13.1.1. 响应结构

响应包含应用程序中使用的所有 Spring Integration 组件, 以及它们之间的链接. 有关该结构的更多信息, 请参见 参考文档.

13.2. 重建 Spring Integration graph

要重新构建 graph ，请向 /actuator/integrationgraph 发出 POST 请求，如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/integrationgraph' -i -X POST
```

这将导致 204 - No Content 响应:

```
HTTP/1.1 204 No Content
```

Chapter 14. Liquibase (liquibase)

liquibase 端点提供有关 Liquibase 应用数据库的变更集信息.

14.1. 检索更改

要检索更改, 请向 /actuator/liquibase 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/liquibase' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 688

{
  "contexts" : {
    "application" : {
      "liquibaseBeans" : {
        "liquibase" : {
          "changeSets" : [ {
            "author" : "marceloverdijk",
            "changeLog" : "classpath:/db/changelog/db.changelog-master.yaml",
            "comments" : "",
            "contexts" : [ ],
            "dateExecuted" : "2021-05-29T07:00:28.436Z",
            "deploymentId" : "2271628396",
            "description" : "createTable tableName=customer",
            "execType" : "EXECUTED",
            "id" : "1",
            "labels" : [ ],
            "checksum" : "8:46debf252cce6d7b25e28ddeb9fc4bf6",
            "orderExecuted" : 1
          } ]
        }
      }
    }
  }
}
```

14.1.1. 响应结构

该响应包含应用程序的 Liquibase 更改集的详细信息. 下表描述了响应的结构:

Path	Type	Description
`contexts`	`Object`	Application contexts keyed by id
`contexts.*.liquibaseBeans.*.changeSets`	`Array`	Change sets made by the Liquibase beans, keyed by bean name.
`contexts.*.liquibaseBeans.*.changeSets[].author`	`String`	Author of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].changeLog`	`String`	Change log that contains the change set.
`contexts.*.liquibaseBeans.*.changeSets[].comments`	`String`	Comments on the change set.
`contexts.*.liquibaseBeans.*.changeSets[].contexts`	`Array`	Contexts of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].dateExecuted`	`String`	Timestamp of when the change set was executed.
`contexts.*.liquibaseBeans.*.changeSets[].deploymentId`	`String`	ID of the deployment that ran the change set.
`contexts.*.liquibaseBeans.*.changeSets[].description`	`String`	Description of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].execType`	`String`	Execution type of the change set (`EXECUTED`, `FAILED`, `SKIPPED`, `RERAN`, `MARK_RAN`).

Path	Type	Description
`contexts.*.liquibaseBeans.*.changeSets[].id`	`String`	ID of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].labels`	`Array`	Labels associated with the change set.
`contexts.*.liquibaseBeans.*.changeSets[].checksum`	`String`	Checksum of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].orderExecuted`	`Number`	Order of the execution of the change set.
`contexts.*.liquibaseBeans.*.changeSets[].tag`	`String`	Tag associated with the change set, if any.
`contexts.*.parentId`	`String`	Id of the parent application context, if any.

Chapter 15. 日志文件 (logfile)

`logfile` 端点可以访问应用程序的日志文件的内容.

15.1. 检索日志文件

要检索日志文件, 请向 `/actuator/logfile` 发出 `GET` 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/logfile' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Type: text/plain;charset=UTF-8
Content-Length: 4723

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::

2017-08-08 17:12:30.910  INFO 19866 --- [           main]
s.f.SampleWebFreeMarkerApplication       : Starting
SampleWebFreeMarkerApplication on host.local with PID 19866
2017-08-08 17:12:30.913  INFO 19866 --- [           main]
s.f.SampleWebFreeMarkerApplication       : No active profile set, falling back
to default profiles: default
2017-08-08 17:12:30.952  INFO 19866 --- [           main]
ConfigServletWebServerApplicationContext : Refreshing
org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApp
licationContext@76b10754: startup date [Tue Aug 08 17:12:30 BST 2017]; root of
context hierarchy
2017-08-08 17:12:31.878  INFO 19866 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080
(http)
2017-08-08 17:12:31.889  INFO 19866 --- [           main]
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
```

```
2017-08-08 17:12:31.890  INFO 19866 --- [           main]
org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache
Tomcat/8.5.16
2017-08-08 17:12:31.978  INFO 19866 --- [ost-startStop-1]
o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded
WebApplicationContext
2017-08-08 17:12:31.978  INFO 19866 --- [ost-startStop-1]
o.s.web.context.ContextLoader            : Root WebApplicationContext:
initialization completed in 1028 ms
2017-08-08 17:12:32.080  INFO 19866 --- [ost-startStop-1]
o.s.b.w.servlet.ServletRegistrationBean  : Mapping servlet: 'dispatcherServlet'
to [/]
2017-08-08 17:12:32.084  INFO 19866 --- [ost-startStop-1]
o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter:
'characterEncodingFilter' to: [/*]
2017-08-08 17:12:32.084  INFO 19866 --- [ost-startStop-1]
o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter:
'hiddenHttpMethodFilter' to: [/*]
2017-08-08 17:12:32.084  INFO 19866 --- [ost-startStop-1]
o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter:
'httpPutFormContentFilter' to: [/*]
2017-08-08 17:12:32.084  INFO 19866 --- [ost-startStop-1]
o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter:
'requestContextFilter' to: [/*]
2017-08-08 17:12:32.349  INFO 19866 --- [           main]
s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice:
org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApp
licationContext@76b10754: startup date [Tue Aug 08 17:12:30 BST 2017]; root of
context hierarchy
2017-08-08 17:12:32.420  INFO 19866 --- [           main]
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public
org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,
java.lang.Object>>
org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.er
ror(javax.servlet.http.HttpServletRequest)
2017-08-08 17:12:32.421  INFO 19866 --- [           main]
s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped
"{[/error],produces=[text/html]}" onto public
org.springframework.web.servlet.ModelAndView
org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.er
rorHtml(javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResp
onse)
2017-08-08 17:12:32.444  INFO 19866 --- [           main]
o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto
handler of type [class
org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-08-08 17:12:32.444  INFO 19866 --- [           main]
o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of
type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
```

```
2017-08-08 17:12:32.471  INFO 19866 --- [           main]
o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico]
onto handler of type [class
org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-08-08 17:12:32.600  INFO 19866 --- [           main]
o.s.w.s.v.f.FreeMarkerConfigurer         : ClassTemplateLoader for Spring macros
added to FreeMarker configuration
2017-08-08 17:12:32.681  INFO 19866 --- [           main]
o.s.j.e.a.AnnotationMBeanExporter        : Registering beans for JMX exposure on
startup
2017-08-08 17:12:32.744  INFO 19866 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080
(http)
2017-08-08 17:12:32.750  INFO 19866 --- [           main]
s.f.SampleWebFreeMarkerApplication       : Started
SampleWebFreeMarkerApplication in 2.172 seconds (JVM running for 2.479)
```

15.2. 检索部分日志文件

使用 Jersey 时，不支持检索部分日志文件.

要检索部分日志文件，请向 `/actuator/logfile` 发送 `GET` 请求并使用 `Range` 头进行请求，如以下基于 `curl` 的示例所示：

```
$ curl 'http://localhost:8080/actuator/logfile' -i -X GET \
    -H 'Range: bytes=0-1023'
```

前面的示例检索日志文件的前 1024 个字节. 产生的响应类似于以下内容：

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes
Content-Type: text/plain;charset=UTF-8
Content-Range: bytes 0-1023/4723
Content-Length: 1024

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::

2017-08-08 17:12:30.910  INFO 19866 --- [           main]
s.f.SampleWebFreeMarkerApplication       : Starting
SampleWebFreeMarkerApplication on host.local with PID 19866
2017-08-08 17:12:30.913  INFO 19866 --- [           main]
s.f.SampleWebFreeMarkerApplication       : No active profile set, falling back
to default profiles: default
2017-08-08 17:12:30.952  INFO 19866 --- [           main]
ConfigServletWebServerApplicationContext : Refreshing
org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApp
licationContext@76b10754: startup date [Tue Aug 08 17:12:30 BST 2017]; root of
context hierarchy
2017-08-08 17:12:31.878  INFO 19866 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(
```

Chapter 16. Loggers (loggers)

loggers 端点可以访问应用程序的记录程序及其级别的配置.

16.1. 检索所有记录器

要检索应用程序的记录器, 请向 /actuator/loggers 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/loggers' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 791

{
  "levels" : [ "OFF", "FATAL", "ERROR", "WARN", "INFO", "DEBUG", "TRACE" ],
  "loggers" : {
    "ROOT" : {
      "configuredLevel" : "INFO",
      "effectiveLevel" : "INFO"
    },
    "com.example" : {
      "configuredLevel" : "DEBUG",
      "effectiveLevel" : "DEBUG"
    }
  },
  "groups" : {
    "test" : {
      "configuredLevel" : "INFO",
      "members" : [ "test.member1", "test.member2" ]
    },
    "web" : {
      "members" : [ "org.springframework.core.codec",
"org.springframework.http", "org.springframework.web",
"org.springframework.boot.actuate.endpoint.web",
"org.springframework.boot.web.servlet.ServletContextInitializerBeans" ]
    },
    "sql" : {
      "members" : [ "org.springframework.jdbc.core", "org.hibernate.SQL",
"org.jooq.tools.LoggerListener" ]
    }
  }
}
```

16.1.1. 响应结构

该响应包含应用程序记录器的详细信息. 下表描述了响应的结构:

Path	Type	Description
`levels`	`Array`	Levels support by the logging system.
`loggers`	`Object`	Loggers keyed by name.

Path	Type	Description
groups	Object	Logger groups keyed by name
loggers.*.configuredLevel	String	Configured level of the logger, if any.
loggers.*.effectiveLevel	String	Effective level of the logger.
groups.*.configuredLevel	String	Configured level of the logger group, if any.
groups.*.members	Array	Loggers that are part of this group

16.2. 检索单个记录器

要检索单个记录器，请向 `/actuator/loggers/{logger.name}` 发出 `GET` 请求，如以下基于 `curl` 的示例所示：

```
$ curl 'http://localhost:8080/actuator/loggers/com.example' -i -X GET
```

前面的示例检索有关名为 `com.example` 的记录器的信息．产生的响应类似于以下内容：

```
HTTP/1.1 200 OK
Content-Disposition: inline;filename=f.txt
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 61

{
  "configuredLevel" : "INFO",
  "effectiveLevel" : "INFO"
}
```

16.2.1. 响应结构

该响应包含所请求记录器的详细信息．下表描述了响应的结构：

Path	Type	Description
configuredLevel	String	Configured level of the logger, if any.
effectiveLevel	String	Effective level of the logger.

16.3. 检索单个组

要检索单个组，请向 /actuator/loggers/{group.name} 发出 GET 请求，如以下基于 curl 的示例所示：

```
$ curl 'http://localhost:8080/actuator/loggers/test' -i -X GET
```

前面的示例检索有关名为 test 的记录器组的信息. 产生的响应类似于以下内容：

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 82

{
  "configuredLevel" : "INFO",
  "members" : [ "test.member1", "test.member2" ]
}
```

16.3.1. 响应结构

响应包含所请求组的详细信息. 下表描述了响应的结构：

Path	Type	Description
configuredLevel	String	Configured level of the logger group, if any.
members	Array	Loggers that are part of this group

16.4. 设置日志级别

要设置记录器的级别，请向 /actuator/loggers/{logger.name} JSON 主体发送 POST 请求，以指定记录器的配置级别，如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/loggers/com.example' -i -X POST \
    -H 'Content-Type: application/json' \
    -d '{"configuredLevel":"debug"}'
```

前面的例子中设置了 configuredLevel 所述的 com.example 记录器 DEBUG.

16.4.1. 请求结构

该请求指定所需的记录器级别. 下表描述了请求的结构:

Path	Type	Description
configuredLevel	String	Level for the logger. May be omitted to clear the level.

16.5. 设置组的日志级别

要设置记录器的级别，请向 /actuator/loggers/{group.name} JSON主体发送 POST 请求，以指定记录器组的配置级别，如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/loggers/test' -i -X POST \
    -H 'Content-Type: application/json' \
    -d '{"configuredLevel":"debug"}'
```

前面的例子中设置了 configuredLevel 所述的 test 记录器组 DEBUG.

16.5.1. 请求结构

该请求指定记录器组的所需级别. 下表描述了请求的结构:

Path	Type	Description
`configuredLevel`	`String`	Level for the logger. May be omitted to clear the level.

16.6. 清除日志级别

要清除记录器的级别，请向 `/actuator/loggers/{logger.name}` 使用包含空对象的 JSON 主体发出 `POST` 请求，如以下基于 `curl` 的示例所示:

```
$ curl 'http://localhost:8080/actuator/loggers/com.example' -i -X POST \
    -H 'Content-Type: application/json' \
    -d '{}'
```

前面的示例清除了 `com.example` 记录器的已配置级别.

Chapter 17. 映射 (mappings)

mappings 端点提供有关应用程序的请求映射的信息.

17.1. 检索映射

要检索映射, 请向 /actuator/mappings 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:34321/actuator/mappings' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Transfer-Encoding: chunked
Date: Sat, 29 May 2021 07:00:30 GMT
Content-Length: 5339

{
  "contexts" : {
    "application" : {
      "mappings" : {
        "dispatcherServlets" : {
          "dispatcherServlet" : [ {
            "handler" : "Actuator root web endpoint",
            "predicate" : "{GET [/actuator], produces [application/vnd.spring-
boot.actuator.v3+json || application/vnd.spring-boot.actuator.v2+json ||
application/json]}",
            "details" : {
              "handlerMethod" : {
                "className" :
"org.springframework.boot.actuate.endpoint.web.servlet.WebMvcEndpointHandlerMapp
ing.WebMvcLinksHandler",
                "name" : "links",
                "descriptor" :
"(Ljavax/servlet/http/HttpServletRequest;Ljavax/servlet/http/HttpServletResponse
;)Ljava/lang/Object;"
              },
              "requestMappingConditions" : {
                "consumes" : [ ],
                "headers" : [ ],
                "methods" : [ "GET" ],
                "params" : [ ],
                "patterns" : [ "/actuator" ],
                "produces" : [ {
                  "mediaType" : "application/vnd.spring-boot.actuator.v3+json",
                  "negated" : false
                }, {
                  "mediaType" : "application/vnd.spring-boot.actuator.v2+json",
                  "negated" : false
                }, {
                  "mediaType" : "application/json",
                  "negated" : false
                } ]
              }
            }
          }, {
            "handler" : "Actuator web endpoint 'mappings'",
            "predicate" : "{GET [/actuator/mappings], produces
[application/vnd.spring-boot.actuator.v3+json || application/vnd.spring-
boot.actuator.v2+json || application/json]}",
            "details" : {
              "handlerMethod" : {
                "className" :
"org.springframework.boot.actuate.endpoint.web.servlet.AbstractWebMvcEndpointHan
dlerMapping.OperationHandler",
                "name" : "handle",
                "descriptor" :
"(Ljavax/servlet/http/HttpServletRequest;Ljava/util/Map;)Ljava/lang/Object;"
              },
              "requestMappingConditions" : {
                "consumes" : [ ],
                "headers" : [ ],
                "methods" : [ "GET" ],
                "params" : [ ],
                "patterns" : [ "/actuator/mappings" ],
                "produces" : [ {
                  "mediaType" : "application/vnd.spring-boot.actuator.v3+json",
                  "negated" : false
                }, {
                  "mediaType" : "application/vnd.spring-boot.actuator.v2+json",
                  "negated" : false
                }, {
                  "mediaType" : "application/json",
                  "negated" : false
                } ]
              }
            }
          }, {
            "handler" :
"org.springframework.boot.actuate.autoconfigure.endpoint.web.documentation.Mappi
```

```
ngsEndpointServletDocumentationTests$ExampleController#example()",
            "predicate" : "{POST [/], params [a!=alpha], headers [X-Custom=Foo],
consumes [application/json || !application/xml], produces [text/plain]}",
            "details" : {
              "handlerMethod" : {
                "className" :
"org.springframework.boot.actuate.autoconfigure.endpoint.web.documentation.Mappi
ngsEndpointServletDocumentationTests.ExampleController",
                "name" : "example",
                "descriptor" : "()Ljava/lang/String;"
              },
              "requestMappingConditions" : {
                "consumes" : [ {
                  "mediaType" : "application/json",
                  "negated" : false
                }, {
                  "mediaType" : "application/xml",
                  "negated" : true
                } ],
                "headers" : [ {
                  "name" : "X-Custom",
                  "value" : "Foo",
                  "negated" : false
                } ],
                "methods" : [ "POST" ],
                "params" : [ {
                  "name" : "a",
                  "value" : "alpha",
                  "negated" : true
                } ],
                "patterns" : [ "/" ],
                "produces" : [ {
                  "mediaType" : "text/plain",
                  "negated" : false
                } ]
              }
            }
          }, {
            "handler" : "ResourceHttpRequestHandler [Classpath [META-
INF/resources/webjars/]]",
            "predicate" : "/webjars/**"
          }, {
            "handler" : "ResourceHttpRequestHandler [Classpath [META-
INF/resources/], Classpath [resources/], Classpath [static/], Classpath
[public/], ServletContext [/]]",
            "predicate" : "/**"
          } ]
        },
        "servletFilters" : [ {
          "servletNameMappings" : [ ],
          "urlPatternMappings" : [ "/*" ],
          "name" : "requestContextFilter",
          "className" :
"org.springframework.boot.web.servlet.filter.OrderedRequestContextFilter"
        }, {
          "servletNameMappings" : [ ],
          "urlPatternMappings" : [ "/*" ],
          "name" : "formContentFilter",
          "className" :
"org.springframework.boot.web.servlet.filter.OrderedFormContentFilter"
        } ],
        "servlets" : [ {
          "mappings" : [ "/" ],
          "name" : "dispatcherServlet",
          "className" : "org.springframework.web.servlet.DispatcherServlet"
        } ]
      }
    }
  }
}
```

17.1.1. 响应结构

该响应包含应用程序映射的详细信息. 响应中找到的项目取决于 Web
应用程序的类型(reactive 或 Servlet 的). 下表描述了响应的常见元素的结构:

Path	Type	Description
`contexts`	`Object`	Application contexts keyed by id.
`contexts.*.mappings`	`Object`	Mappings in the context, keyed by mapping type.
`contexts.*.mappings.dispatcherServlets`	`Object`	Dispatcher servlet mappings, if any.
`contexts.*.mappings.servletFilters`	`Array`	Servlet filter mappings, if any.
`contexts.*.mappings.servlets`	`Array`	Servlet mappings, if any.

Path	Type	Description
`contexts.*.mappings.dispatcherHandlers`	`Object`	Dispatcher handler mappings, if any.
`contexts.*.parentId`	`String`	Id of the parent application context, if any.

`contexts.*.mappings` 以下各节介绍了可能在其中找到的条目.

17.1.2. Dispatcher Servlets 响应结构

使用 Spring MVC 时, 响应中包含任何 `DispatcherServlet` 请求映射的详细信息 `contexts.*.mappings.dispatcherServlets`. 下表描述了此部分响应的结构:

Path	Type	Description
`*`	`Array`	Dispatcher servlet mappings, if any, keyed by dispatcher servlet bean name.
`*.[].details`	`Object`	Additional implementation-specific details about the mapping. Optional.
`*.[].handler`	`String`	Handler for the mapping.
`*.[].predicate`	`String`	Predicate for the mapping.
`*.[].details.handlerMethod`	`Object`	Details of the method, if any, that will handle requests to this mapping.

Path	Type	Description
*.[].details.handlerMethod.className	Varies	Fully qualified name of the class of the method.
*.[].details.handlerMethod.name	Varies	Name of the method.
*.[].details.handlerMethod.descriptor	Varies	Descriptor of the method as specified in the Java Language Specification.
*.[].details.requestMappingConditions	Object	Details of the request mapping conditions.
*.[].details.requestMappingConditions.consumes	Varies	Details of the consumes condition
*.[].details.requestMappingConditions.consumes.[].mediaType	Varies	Consumed media type.
*.[].details.requestMappingConditions.consumes.[].negated	Varies	Whether the media type is negated.
*.[].details.requestMappingConditions.headers	Varies	Details of the headers condition.
*.[].details.requestMappingConditions.headers.[].name	Varies	Name of the header.
*.[].details.requestMappingConditions.headers.[].value	Varies	Required value of the header, if any.
*.[].details.requestMappingConditions.headers.[].negated	Varies	Whether the value is negated.
*.[].details.requestMappingConditions.methods	Varies	HTTP methods that are handled.
*.[].details.requestMappingConditions.params	Varies	Details of the params condition.

Path	Type	Description
`*.[].details.requestMappingConditions.params.[].name`	`Varies`	Name of the parameter.
`*.[].details.requestMappingConditions.params.[].value`	`Varies`	Required value of the parameter, if any.
`*.[].details.requestMappingConditions.params.[].negated`	`Varies`	Whether the value is negated.
`*.[].details.requestMappingConditions.patterns`	`Varies`	Patterns identifying the paths handled by the mapping.
`*.[].details.requestMappingConditions.produces`	`Varies`	Details of the produces condition.
`*.[].details.requestMappingConditions.produces.[].mediaType`	`Varies`	Produced media type.
`*.[].details.requestMappingConditions.produces.[].negated`	`Varies`	Whether the media type is negated.

17.1.3. Servlets 响应结构

使用 Servlet 技术栈时，响应中包含 Servlet 下方任何映射的详细信息 `contexts.*.mappings.servlets`。下表描述了此部分响应的结构：

Path	Type	Description
`[].mappings`	`Array`	Mappings of the servlet.
`[].name`	`String`	Name of the servlet.
`[].className`	`String`	Class name of the servlet

17.1.4. Servlet Filters 响应结构

使用 Servlet 技术栈时，响应中包含 `Filter` 下方任何映射的详细信息 `contexts.*.mappings.servletFilters`。下表描述了此部分响应的结构：

Path	Type	Description
[].servletNameMappings	Array	Names of the servlets to which the filter is mapped.
[].urlPatternMappings	Array	URL pattern to which the filter is mapped.
[].name	String	Name of the filter.
[].className	String	Class name of the filter

17.1.5. Dispatcher Handlers 响应结构

当使用 Spring WebFlux 时, 响应 DispatcherHandler
在下面包含任何请求映射的详细信息 contexts.*.mappings.dispatcherHandlers.
下表描述了此部分响应的结构:

Path	Type	Description
*	Array	Dispatcher handler mappings, if any, keyed by dispatcher handler bean name.
*.[].details	Object	Additional implementation-specific details about the mapping. Optional.
*.[].handler	String	Handler for the mapping.
*.[].predicate	String	Predicate for the mapping.
*.[].details.requestMappingConditions	Object	Details of the request mapping conditions.

Path	Type	Description
`*.[].details.requestMappingConditions.consumes`	`Array`	Details of the consumes condition
`*.[].details.requestMappingConditions.consumes.[].mediaType`	`String`	Consumed media type.
`*.[].details.requestMappingConditions.consumes.[].negated`	`Boolean`	Whether the media type is negated.
`*.[].details.requestMappingConditions.headers`	`Array`	Details of the headers condition.
`*.[].details.requestMappingConditions.headers.[].name`	`String`	Name of the header.
`*.[].details.requestMappingConditions.headers.[].value`	`String`	Required value of the header, if any.
`*.[].details.requestMappingConditions.headers.[].negated`	`Boolean`	Whether the value is negated.
`*.[].details.requestMappingConditions.methods`	`Array`	HTTP methods that are handled.
`*.[].details.requestMappingConditions.params`	`Array`	Details of the params condition.
`*.[].details.requestMappingConditions.params.[].name`	`String`	Name of the parameter.
`*.[].details.requestMappingConditions.params.[].value`	`String`	Required value of the parameter, if any.
`*.[].details.requestMappingConditions.params.[].negated`	`Boolean`	Whether the value is negated.
`*.[].details.requestMappingConditions.patterns`	`Array`	Patterns identifying the paths handled by the mapping.

Path	Type	Description
`*.[].details.requestMappingConditions.produces`	`Array`	Details of the produces condition.
`*.[].details.requestMappingConditions.produces.[].mediaType`	`String`	Produced media type.
`*.[].details.requestMappingConditions.produces.[].negated`	`Boolean`	Whether the media type is negated.
`*.[].details.handlerMethod`	`Object`	Details of the method, if any, that will handle requests to this mapping.
`*.[].details.handlerMethod.className`	`String`	Fully qualified name of the class of the method.
`*.[].details.handlerMethod.name`	`String`	Name of the method.
`*.[].details.handlerMethod.descriptor`	`String`	Descriptor of the method as specified in the Java Language Specification.
`*.[].details.handlerFunction`	`Object`	Details of the function, if any, that will handle requests to this mapping.
`*.[].details.handlerFunction.className`	`String`	Fully qualified name of the class of the function.

Chapter 18. 指标 (metrics)

metrics 端点可以访问应用程序指标.

18.1. 检索指标名称

要检索可用指标的名称, 请向 /actuator/metrics 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/metrics' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 154

{
  "names" : [ "jvm.buffer.count", "jvm.buffer.memory.used",
"jvm.buffer.total.capacity", "jvm.memory.committed", "jvm.memory.max",
"jvm.memory.used" ]
}
```

18.1.1. 响应结构

该响应包含指标名称的详细信息. 下表描述了响应的结构:

Path	Type	Description
names	Array	Names of the known metrics.

18.2. 检索指标

要检索指标, 请向 /actuator/metrics/{metric.name} 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/metrics/jvm.memory.max' -i -X GET
```

前面的示例检索有关名为 `jvm.memory.max` 的信息 . 产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Disposition: inline;filename=f.txt
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 474

{
  "name" : "jvm.memory.max",
  "description" : "The maximum amount of memory in bytes that can be used for
memory management",
  "baseUnit" : "bytes",
  "measurements" : [ {
    "statistic" : "VALUE",
    "value" : 2.368733183E9
  } ],
  "availableTags" : [ {
    "tag" : "area",
    "values" : [ "heap", "nonheap" ]
  }, {
    "tag" : "id",
    "values" : [ "Compressed Class Space", "PS Old Gen", "PS Survivor Space",
"Metaspace", "PS Eden Space", "Code Cache" ]
  } ]
}
```

18.2.1. 查询参数

端点使用查询参数通过其标签 drill down 到指标. 下表显示了单个受支持的查询参数:

Parameter	Description
`tag`	A tag to use for drill-down in the form `name:value`.

18.2.2. 响应结构

响应包含指标标准的详细信息. 下表描述了响应的结构:

Path	Type	Description
`name`	`String`	Name of the metric
`description`	`String`	Description of the metric
`baseUnit`	`String`	Base unit of the metric
`measurements`	`Array`	Measurements of the metric
`measurements[].statistic`	`String`	Statistic of the measurement. (`TOTAL`, `TOTAL_TIME`, `COUNT`, `MAX`, `VALUE`, `UNKNOWN`, `ACTIVE_TASKS`, `DURATION`).
`measurements[].value`	`Number`	Value of the measurement.
`availableTags`	`Array`	Tags that are available for drill-down.
`availableTags[].tag`	`String`	Name of the tag.
`availableTags[].values`	`Array`	Possible values of the tag.

18.3. Drilling Down

要深入了解指标，请向 `/actuator/metrics/{metric.name}` 发送 `GET` 请求并使用 `tag` 查询参数，如以下基于 `curl` 的示例所示：

```
$ curl
'http://localhost:8080/actuator/metrics/jvm.memory.max?tag=area%3Anonheap&tag=id
%3ACompressed+Class+Space' -i -X GET
```

前述示例检索 `jvm.memory.max` 指标，其中该 `area` 标签具有值 `nonheap` 和 `id` 属性具有值 `Compressed Class Space`。产生的响应类似于以下内容：

```
HTTP/1.1 200 OK
Content-Disposition: inline;filename=f.txt
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 263

{
  "name" : "jvm.memory.max",
  "description" : "The maximum amount of memory in bytes that can be used for
memory management",
  "baseUnit" : "bytes",
  "measurements" : [ {
    "statistic" : "VALUE",
    "value" : 1.073741824E9
  } ],
  "availableTags" : [ ]
}
```

Chapter 19. Prometheus (prometheus)

prometheus 端点提供了由 Prometheus 服务器所需的格式 Spring 启动应用程序的指标.

19.1. 检索所有指标

要检索所有指标, 请向 /actuator/prometheus 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/prometheus' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: text/plain;version=0.0.4;charset=utf-8
Content-Length: 2371

# HELP jvm_buffer_total_capacity_bytes An estimate of the total capacity of the
buffers in this pool
# TYPE jvm_buffer_total_capacity_bytes gauge
jvm_buffer_total_capacity_bytes{id="direct",} 262351.0
jvm_buffer_total_capacity_bytes{id="mapped",} 0.0
# HELP jvm_buffer_memory_used_bytes An estimate of the memory that the Java
virtual machine is using for this buffer pool
# TYPE jvm_buffer_memory_used_bytes gauge
jvm_buffer_memory_used_bytes{id="direct",} 262352.0
jvm_buffer_memory_used_bytes{id="mapped",} 0.0
# HELP jvm_buffer_count_buffers An estimate of the number of buffers in the pool
# TYPE jvm_buffer_count_buffers gauge
jvm_buffer_count_buffers{id="direct",} 11.0
jvm_buffer_count_buffers{id="mapped",} 0.0
# HELP jvm_memory_used_bytes The amount of used memory
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes{area="heap",id="PS Survivor Space",} 2.259308E7
jvm_memory_used_bytes{area="heap",id="PS Old Gen",} 1.21200008E8
jvm_memory_used_bytes{area="heap",id="PS Eden Space",} 1.41837928E8
jvm_memory_used_bytes{area="nonheap",id="Metaspace",} 1.09621232E8
jvm_memory_used_bytes{area="nonheap",id="Code Cache",} 3.5761728E7
jvm_memory_used_bytes{area="nonheap",id="Compressed Class Space",} 1.5371952E7
# HELP jvm_memory_committed_bytes The amount of memory in bytes that is
committed for the Java virtual machine to use
# TYPE jvm_memory_committed_bytes gauge
jvm_memory_committed_bytes{area="heap",id="PS Survivor Space",} 2.3068672E7
jvm_memory_committed_bytes{area="heap",id="PS Old Gen",} 1.76160768E8
jvm_memory_committed_bytes{area="heap",id="PS Eden Space",} 2.96747008E8
jvm_memory_committed_bytes{area="nonheap",id="Metaspace",} 1.17989376E8
jvm_memory_committed_bytes{area="nonheap",id="Code Cache",} 3.76832E7
jvm_memory_committed_bytes{area="nonheap",id="Compressed Class Space",}
1.6826368E7
# HELP jvm_memory_max_bytes The maximum amount of memory in bytes that can be
used for memory management
# TYPE jvm_memory_max_bytes gauge
jvm_memory_max_bytes{area="heap",id="PS Survivor Space",} 2.3068672E7
jvm_memory_max_bytes{area="heap",id="PS Old Gen",} 7.16177408E8
jvm_memory_max_bytes{area="heap",id="PS Eden Space",} 3.014656E8
jvm_memory_max_bytes{area="nonheap",id="Metaspace",} -1.0
jvm_memory_max_bytes{area="nonheap",id="Code Cache",} 2.5165824E8
jvm_memory_max_bytes{area="nonheap",id="Compressed Class Space",} 1.073741824E9
```

19.1.1. 查询参数

endpoint 使用查询参数来限制它返回的 samples. 支持以下的查询参数:

Parameter	Description
includedNames	Restricts the samples to those that match the names. Optional.

19.2. 检索过滤的指标

要检索与特定名称匹配的指标， 请使用 includedNames 查询参数向 /actuator/prometheus 发出 GET 请求， 如以下基于 curl 的示例所示:

```
$ curl
'http://localhost:8080/actuator/prometheus?includedNames=jvm_memory_used_bytes%2
Cjvm_memory_committed_bytes' -i -X GET
```

得到的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: text/plain;version=0.0.4;charset=utf-8
Content-Length: 1108

# HELP jvm_memory_used_bytes The amount of used memory
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes{area="heap",id="PS Survivor Space",} 2.259308E7
jvm_memory_used_bytes{area="heap",id="PS Old Gen",} 1.21200008E8
jvm_memory_used_bytes{area="heap",id="PS Eden Space",} 1.48286728E8
jvm_memory_used_bytes{area="nonheap",id="Metaspace",} 1.09624096E8
jvm_memory_used_bytes{area="nonheap",id="Code Cache",} 3.5773952E7
jvm_memory_used_bytes{area="nonheap",id="Compressed Class Space",} 1.537252E7
# HELP jvm_memory_committed_bytes The amount of memory in bytes that is
committed for the Java virtual machine to use
# TYPE jvm_memory_committed_bytes gauge
jvm_memory_committed_bytes{area="heap",id="PS Survivor Space",} 2.3068672E7
jvm_memory_committed_bytes{area="heap",id="PS Old Gen",} 1.76160768E8
jvm_memory_committed_bytes{area="heap",id="PS Eden Space",} 2.96747008E8
jvm_memory_committed_bytes{area="nonheap",id="Metaspace",} 1.17989376E8
jvm_memory_committed_bytes{area="nonheap",id="Code Cache",} 3.76832E7
jvm_memory_committed_bytes{area="nonheap",id="Compressed Class Space",}
1.6826368E7
```

Chapter 20. 定时任务 (scheduledtasks)

scheduledtasks 端点提供有关应用程序的定时任务的信息. .

20.1. 检索定时任务

要检索计划的任务, 请向 /actuator/scheduledtasks 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/scheduledtasks' -i -X GET
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 629

{
  "cron" : [ {
    "runnable" : {
      "target" : "com.example.Processor.processOrders"
    },
    "expression" : "0 0 0/3 1/1 * ?"
  } ],
  "fixedDelay" : [ {
    "runnable" : {
      "target" : "com.example.Processor.purge"
    },
    "initialDelay" : 5000,
    "interval" : 5000
  } ],
  "fixedRate" : [ {
    "runnable" : {
      "target" : "com.example.Processor.retrieveIssues"
    },
    "initialDelay" : 10000,
    "interval" : 3000
  } ],
  "custom" : [ {
    "runnable" : {
      "target" : "com.example.Processor$CustomTriggeredRunnable"
    },
    "trigger" : "com.example.Processor$CustomTrigger@31b08638"
  } ]
}
```

20.1.1. 响应结构

该响应包含应用程序计划任务的详细信息. 下表描述了响应的结构:

Path	Type	Description
cron	Array	Cron tasks, if any.
cron.[].runnable.target	String	Target that will be executed.
cron.[].expression	String	Cron expression.
fixedDelay	Array	Fixed delay tasks, if any.

Path	Type	Description
`fixedDelay.[].runnable.target`	`String`	Target that will be executed.
`fixedDelay.[].initialDelay`	`Number`	Delay, in milliseconds, before first execution.
`fixedDelay.[].interval`	`Number`	Interval, in milliseconds, between the end of the last execution and the start of the next.
`fixedRate`	`Array`	Fixed rate tasks, if any.
`fixedRate.[].runnable.target`	`String`	Target that will be executed.
`fixedRate.[].interval`	`Number`	Interval, in milliseconds, between the start of each execution.
`fixedRate.[].initialDelay`	`Number`	Delay, in milliseconds, before first execution.
`custom`	`Array`	Tasks with custom triggers, if any.
`custom.[].runnable.target`	`String`	Target that will be executed.
`custom.[].trigger`	`String`	Trigger for the task.

Chapter 21. Sessions (sessions)

sessions 端点提供由 Spring Session 管理的应用程序的 HTTP session

21.1. 检索会话

要检索会话, 请向 /actuator/sessions 发出 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/sessions?username=alice' -i -X GET
```

前面的示例为 检索用户名为 alice 的用户所有会话. 产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 753

{
  "sessions" : [ {
    "id" : "6251867e-f8b0-4477-baf0-9823d6c0910f",
    "attributeNames" : [ ],
    "creationTime" : "2021-05-29T05:00:32.433Z",
    "lastAccessedTime" : "2021-05-29T07:00:20.433Z",
    "maxInactiveInterval" : 1800,
    "expired" : false
  }, {
    "id" : "bad3de2a-28c6-45f6-a5b3-32e215a4578e",
    "attributeNames" : [ ],
    "creationTime" : "2021-05-28T19:00:32.432Z",
    "lastAccessedTime" : "2021-05-29T06:59:47.432Z",
    "maxInactiveInterval" : 1800,
    "expired" : false
  }, {
    "id" : "4db5efcc-99cb-4d05-a52c-b49acfbb7ea9",
    "attributeNames" : [ ],
    "creationTime" : "2021-05-29T02:00:32.433Z",
    "lastAccessedTime" : "2021-05-29T06:59:55.433Z",
    "maxInactiveInterval" : 1800,
    "expired" : false
  } ]
}
```

21.1.1. 查询参数

端点使用查询参数来限制其返回的会话. 下表显示了单个必需的查询参数:

Parameter	Description
`username`	Name of the user.

21.1.2. 响应结构

响应包含匹配会话的详细信息. 下表描述了响应的结构:

Path	Type	Description
`sessions`	`Array`	Sessions for the given username.
`sessions.[].id`	`String`	ID of the session.
`sessions.[].attributeNames`	`Array`	Names of the attributes stored in the session.
`sessions.[].creationTime`	`String`	Timestamp of when the session was created.
`sessions.[].lastAccessedTime`	`String`	Timestamp of when the session was last accessed.
`sessions.[].maxInactiveInterval`	`Number`	Maximum permitted period of inactivity, in seconds, before the session will expire.
`sessions.[].expired`	`Boolean`	Whether the session has expired.

21.2. 检索单个会话

要检索单个会话, 请向 `/actuator/sessions/{id}` 发出 `GET` 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/sessions/4db5efcc-99cb-4d05-a52c-
b49acfbb7ea9' -i -X GET
```

前面的例子中检索 `id` 为 `4db5efcc-99cb-4d05-a52c-b49acfbb7ea9` 的 session. 产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 228

{
  "id" : "4db5efcc-99cb-4d05-a52c-b49acfbb7ea9",
  "attributeNames" : [ ],
  "creationTime" : "2021-05-29T02:00:32.433Z",
  "lastAccessedTime" : "2021-05-29T06:59:55.433Z",
  "maxInactiveInterval" : 1800,
  "expired" : false
}
```

21.2.1. 响应结构

响应包含请求的会话的详细信息. 下表描述了响应的结构:

Path	Type	Description
`id`	`String`	ID of the session.
`attributeNames`	`Array`	Names of the attributes stored in the session.
`creationTime`	`String`	Timestamp of when the session was created.
`lastAccessedTime`	`String`	Timestamp of when the session was last accessed.
`maxInactiveInterval`	`Number`	Maximum permitted period of inactivity, in seconds, before the session will expire.

Path	Type	Description
expired	Boolean	Whether the session has expired.

21.3. 删除会话

要删除会话，请向 /actuator/sessions/{id} 发出 DELETE 请求，如以下基于 curl 的示例所示：

```
$ curl 'http://localhost:8080/actuator/sessions/4db5efcc-99cb-4d05-a52c-
b49acfbb7ea9' -i -X DELETE
```

删除 id 为 4db5efcc-99cb-4d05-a52c-b49acfbb7ea9 的 session

Chapter 22. Shutdown (shutdown)

shutdown 端点被用来关闭应用程序.

22.1. 关闭应用程序

要关闭应用程序, 请向 /actuator/shutdown 发出 POST 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/shutdown' -i -X POST
```

产生类似于以下内容的响应:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 41

{
  "message" : "Shutting down, bye..."
}
```

22.1.1. 响应结构

该响应包含关闭请求结果的详细信息. 下表描述了响应的结构:

Path	Type	Description
message	String	Message describing the result of the request.

Chapter 23. Application Startup (`startup`)

`startup` 端点提供有关应用程序启动顺序的信息.

23.1. 检索应用程序启动顺序

要在应用程序启动阶段返回迄今为止记录的步骤, 请向 `/actuator/startup` 发出 `POST` 请求, 如下面基于 `curl` 的示例所示:

```
$ curl 'http://localhost:8080/actuator/startup' -i -X POST
```

得到的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.spring-boot.actuator.v3+json
Content-Length: 905

{
  "springBootVersion" : "2.4.5",
  "timeline" : {
    "startTime" : "2021-05-29T07:00:33.681Z",
    "events" : [ {
      "startupStep" : {
        "name" : "spring.boot.application.starting",
        "id" : 1,
        "parentId" : 0,
        "tags" : [ {
          "key" : "mainApplicationClass",
          "value" : "com.example.startup.StartupApplication"
        } ]
      },
      "startTime" : "2021-05-29T07:00:33.803098682Z",
      "endTime" : "2021-05-29T07:00:33.803522683Z",
      "duration" : "PT0.000424001S"
    }, {
      "startupStep" : {
        "name" : "spring.beans.instantiate",
        "id" : 2,
        "parentId" : 0,
        "tags" : [ {
          "key" : "beanName",
          "value" : "homeController"
        } ]
      },
      "startTime" : "2021-05-29T07:00:33.803571083Z",
      "endTime" : "2021-05-29T07:00:33.803578883Z",
      "duration" : "PT0.0000078S"
    } ]
  }
}
```

每个这样的调用都会从缓冲区中删除返回的步骤:

23.1.1. 响应结构

响应包含应用程序启动步骤的详细信息. 下表描述了响应的结构:

Path	Type	Description
springBootVersion	String	Spring Boot version for this application.
timeline.startTime	String	Start time of the application.
timeline.events	Array	An array of steps collected during application startup so far.
timeline.events.[].startTime	String	The timestamp of the start of this event.
timeline.events.[].endTime	String	The timestamp of the end of this event.
timeline.events.[].duration	String	The precise duration of this event.
timeline.events.[].startupStep.name	String	The name of the StartupStep.
timeline.events.[].startupStep.id	Number	The id of this StartupStep.
timeline.events.[].startupStep.parentId	Number	The parent id for this StartupStep.
timeline.events.[].startupStep.tags	Array	An array of key/value pairs with additional step info.
timeline.events.[].startupStep.tags[].key	String	The key of the StartupStep Tag.
timeline.events.[].startupStep.tags[].value	String	The value of the StartupStep Tag.

Chapter 24. Thread Dump (threaddump)

threaddump 端点从应用程序的 JVM 提供了一个线程转储.

24.1. 以 JSON 检索线程转储

要将线程转储作为 JSON 检索, 请向 /actuator/threaddump 使用适当的 Accept 头进行 GET 请求, 如以下基于 curl 的示例所示:

```
$ curl 'http://localhost:8080/actuator/threaddump' -i -X GET \
    -H 'Accept: application/json'
```

产生的响应类似于以下内容:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 6357

{
  "threads" : [ {
    "threadName" : "Thread-60",
    "threadId" : 357,
    "blockedTime" : -1,
    "blockedCount" : 0,
    "waitedTime" : -1,
    "waitedCount" : 0,
    "lockOwnerId" : -1,
    "inNative" : false,
    "suspended" : false,
    "threadState" : "RUNNABLE",
    "stackTrace" : [ ],
    "lockedMonitors" : [ ],
    "lockedSynchronizers" : [ ]
  }, {
    "threadName" : "Thread-58",
    "threadId" : 354,
    "blockedTime" : -1,
    "blockedCount" : 0,
    "waitedTime" : -1,
    "waitedCount" : 1,
    "lockOwnerId" : -1,
    "inNative" : false,
    "suspended" : false,
    "threadState" : "TIMED_WAITING",
    "stackTrace" : [ {
      "methodName" : "sleep",
      "fileName" : "Thread.java",
      "lineNumber" : -2,
      "className" : "java.lang.Thread",
      "nativeMethod" : true
    }, {
      "methodName" : "performShutdown",
      "fileName" : "ShutdownEndpoint.java",
      "lineNumber" : 65,
      "className" : "org.springframework.boot.actuate.context.ShutdownEndpoint",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "lineNumber" : -1,
      "className" :
"org.springframework.boot.actuate.context.ShutdownEndpoint$$Lambda$2403/25575452
9",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "Thread.java",
      "lineNumber" : 748,
      "className" : "java.lang.Thread",
      "nativeMethod" : false
    } ],
    "lockedMonitors" : [ ],
    "lockedSynchronizers" : [ ]
  }, {
    "threadName" : "pool-8-thread-1",
    "threadId" : 346,
    "blockedTime" : -1,
    "blockedCount" : 0,
    "waitedTime" : -1,
    "waitedCount" : 0,
    "lockOwnerId" : -1,
    "inNative" : false,
    "suspended" : false,
    "threadState" : "RUNNABLE",
    "stackTrace" : [ {
      "methodName" : "siftUp",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 886,
      "className" :
```

```
"java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue",
      "nativeMethod" : false
    }, {
      "methodName" : "offer",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 1020,
      "className" :
"java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue",
      "nativeMethod" : false
    }, {
      "methodName" : "add",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 1037,
      "className" :
"java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue",
      "nativeMethod" : false
    }, {
      "methodName" : "add",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 809,
      "className" :
"java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue",
      "nativeMethod" : false
    }, {
      "methodName" : "delayedExecute",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 328,
      "className" : "java.util.concurrent.ScheduledThreadPoolExecutor",
      "nativeMethod" : false
    }, {
      "methodName" : "schedule",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 533,
      "className" : "java.util.concurrent.ScheduledThreadPoolExecutor",
      "nativeMethod" : false
    }, {
      "methodName" : "schedule",
      "fileName" : "Executors.java",
      "lineNumber" : 729,
      "className" :
"java.util.concurrent.Executors$DelegatedScheduledExecutorService",
      "nativeMethod" : false
    }, {
      "methodName" : "schedule",
      "fileName" : "ReschedulingRunnable.java",
      "lineNumber" : 82,
      "className" :
"org.springframework.scheduling.concurrent.ReschedulingRunnable",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "ReschedulingRunnable.java",
      "lineNumber" : 101,
      "className" :
"org.springframework.scheduling.concurrent.ReschedulingRunnable",
      "nativeMethod" : false
    }, {
      "methodName" : "call",
      "fileName" : "Executors.java",
      "lineNumber" : 511,
      "className" : "java.util.concurrent.Executors$RunnableAdapter",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "FutureTask.java",
      "lineNumber" : 266,
      "className" : "java.util.concurrent.FutureTask",
      "nativeMethod" : false
    }, {
      "methodName" : "access$201",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 180,
      "className" :
"java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "ScheduledThreadPoolExecutor.java",
      "lineNumber" : 293,
      "className" :
"java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask",
      "nativeMethod" : false
    }, {
      "methodName" : "runWorker",
      "fileName" : "ThreadPoolExecutor.java",
      "lineNumber" : 1149,
      "className" : "java.util.concurrent.ThreadPoolExecutor",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "ThreadPoolExecutor.java",
      "lineNumber" : 624,
      "className" : "java.util.concurrent.ThreadPoolExecutor$Worker",
      "nativeMethod" : false
    }, {
      "methodName" : "run",
      "fileName" : "Thread.java",
      "lineNumber" : 748,
      "className" : "java.lang.Thread",
      "nativeMethod" : false
    } ],
    "lockedMonitors" : [ {
      "className" : "java.lang.Object",
      "identityHashCode" : 1927829486,
      "lockedStackDepth" : 7,
      "lockedStackFrame" : {
        "methodName" : "schedule",
        "fileName" : "ReschedulingRunnable.java",
        "lineNumber" : 82,
        "className" :
"org.springframework.scheduling.concurrent.ReschedulingRunnable",
        "nativeMethod" : false
      }
    }, {
      "className" : "java.lang.Object",
      "identityHashCode" : 1927829486,
      "lockedStackDepth" : 8,
      "lockedStackFrame" : {
        "methodName" : "run",
        "fileName" : "ReschedulingRunnable.java",
        "lineNumber" : 101,
        "className" :
"org.springframework.scheduling.concurrent.ReschedulingRunnable",
        "nativeMethod" : false
      }
    } ],
    "lockedSynchronizers" : [ {
      "className" : "java.util.concurrent.ThreadPoolExecutor$Worker",
      "identityHashCode" : 1386452798
    }, {
      "className" : "java.util.concurrent.locks.ReentrantLock$NonfairSync",
      "identityHashCode" : 2108179052
    } ]
  } ]
}
```

24.1.1. 响应结构

该响应包含 JVM 线程的详细信息. 下表描述了响应的结构:

Path	Type	Description
`threads`	`Array`	JVM’s threads.
`threads.[].blockedCount`	`Number`	Total number of times that the thread has been blocked.
`threads.[].blockedTime`	`Number`	Time in milliseconds that the thread has spent blocked. -1 if thread contention monitoring is disabled.
`threads.[].daemon`	`Boolean`	Whether the thread is a daemon thread. Only available on Java 9 or later.
`threads.[].inNative`	`Boolean`	Whether the thread is executing native code.
`threads.[].lockName`	`String`	Description of the object on which the thread is blocked, if any.
`threads.[].lockInfo`	`Object`	Object for which the thread is blocked waiting.
`threads.[].lockInfo.className`	`String`	Fully qualified class name of the lock object.
`threads.[].lockInfo.identityHashCode`	`Number`	Identity hash code of the lock object.
`threads.[].lockedMonitors`	`Array`	Monitors locked by this thread, if any
`threads.[].lockedMonitors.[].className`	`String`	Class name of the lock object.

Path	Type	Description
threads.[].lockedMonitors.[].identityHashCode	Number	Identity hash code of the lock object.
threads.[].lockedMonitors.[].lockedStackDepth	Number	Stack depth where the monitor was locked.
threads.[].lockedMonitors.[].lockedStackFrame	Object	Stack frame that locked the monitor.
threads.[].lockedSynchronizers	Array	Synchronizers locked by this thread.
threads.[].lockedSynchronizers.[].className	String	Class name of the locked synchronizer.
threads.[].lockedSynchronizers.[].identityHashCode	Number	Identity hash code of the locked synchronizer.
threads.[].lockOwnerId	Number	ID of the thread that owns the object on which the thread is blocked. -1 if the thread is not blocked.
threads.[].lockOwnerName	String	Name of the thread that owns the object on which the thread is blocked, if any.
threads.[].priority	Number	Priority of the thread. Only available on Java 9 or later.
threads.[].stackTrace	Array	Stack trace of the thread.

Path	Type	Description
threads.[].stackTrace.[].classLoaderName	String	Name of the class loader of the class that contains the execution point identified by this entry, if any. Only available on Java 9 or later.
threads.[].stackTrace.[].className	String	Name of the class that contains the execution point identified by this entry.
threads.[].stackTrace.[].fileName	String	Name of the source file that contains the execution point identified by this entry, if any.
threads.[].stackTrace.[].lineNumber	Number	Line number of the execution point identified by this entry. Negative if unknown.
threads.[].stackTrace.[].methodName	String	Name of the method.
threads.[].stackTrace.[].moduleName	String	Name of the module that contains the execution point identified by this entry, if any. Only available on Java 9 or later.

Path	Type	Description
`threads.[].stackTrace.[].moduleVersion`	`String`	Version of the module that contains the execution point identified by this entry, if any. Only available on Java 9 or later.
`threads.[].stackTrace.[].nativeMethod`	`Boolean`	Whether the execution point is a native method.
`threads.[].suspended`	`Boolean`	Whether the thread is suspended.
`threads.[].threadId`	`Number`	ID of the thread.
`threads.[].threadName`	`String`	Name of the thread.
`threads.[].threadState`	`String`	State of the thread (`NEW`, `RUNNABLE`, `BLOCKED`, `WAITING`, `TIMED_WAITING`, `TERMINATED`).
`threads.[].waitedCount`	`Number`	Total number of times that the thread has waited for notification.
`threads.[].waitedTime`	`Number`	Time in milliseconds that the thread has spent waiting. -1 if thread contention monitoring is disabled

24.2. 以文本形式检索线程转储

要以文本形式检索线程转储，请向 `/actuator/threaddump` 发送 `GET` 请求 `Accept` 头为

`text/plain`，如以下基于 `curl` 的示例所示：

```
$ curl 'http://localhost:8080/actuator/threaddump' -i -X GET \
    -H 'Accept: text/plain'
```

产生的响应类似于以下内容：

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=UTF-8
Content-Length: 44581

2021-05-29 15:00:34
Full thread dump OpenJDK 64-Bit Server VM (25.292-b10 mixed mode):

"Thread-58" - Thread t@354
   java.lang.Thread.State: TIMED_WAITING
    at java.lang.Thread.sleep(Native Method)
    at
org.springframework.boot.actuate.context.ShutdownEndpoint.performShutdown(Shutdo
wnEndpoint.java:65)
    at
org.springframework.boot.actuate.context.ShutdownEndpoint$$Lambda$2403/255754529
.run(Unknown Source)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"pool-8-thread-1" - Thread t@346
   java.lang.Thread.State: RUNNABLE
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.siftUp(Schedul
edThreadPoolExecutor.java:886)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.offer(Schedule
dThreadPoolExecutor.java:1020)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.add(ScheduledT
hreadPoolExecutor.java:1037)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.add(ScheduledT
hreadPoolExecutor.java:809)
    at
java.util.concurrent.ScheduledThreadPoolExecutor.delayedExecute(ScheduledThreadP
oolExecutor.java:328)
```

```
    at
java.util.concurrent.ScheduledThreadPoolExecutor.schedule(ScheduledThreadPoolExe
cutor.java:533)
    at
java.util.concurrent.Executors$DelegatedScheduledExecutorService.schedule(Execut
ors.java:729)
    at
org.springframework.scheduling.concurrent.ReschedulingRunnable.schedule(Reschedu
lingRunnable.java:82)
    - locked <72e857ee> (a java.lang.Object)
    at
org.springframework.scheduling.concurrent.ReschedulingRunnable.run(ReschedulingR
unnable.java:101)
    - locked <72e857ee> (a java.lang.Object)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(
ScheduledThreadPoolExecutor.java:180)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Schedul
edThreadPoolExecutor.java:293)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - Locked <52a3973e> (a java.util.concurrent.ThreadPoolExecutor$Worker)
    - Locked <7da8426c> (a java.util.concurrent.locks.ReentrantLock$NonfairSync)

"http-nio-auto-14-Acceptor" - Thread t@339
   java.lang.Thread.State: RUNNABLE
    at sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
    at
sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:421)
    at
sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:249)
    - locked <36d282e6> (a java.lang.Object)
    at
org.apache.tomcat.util.net.NioEndpoint.serverSocketAccept(NioEndpoint.java:574)
    at
org.apache.tomcat.util.net.NioEndpoint.serverSocketAccept(NioEndpoint.java:80)
    at org.apache.tomcat.util.net.Acceptor.run(Acceptor.java:106)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-ClientPoller" - Thread t@338
   java.lang.Thread.State: RUNNABLE
    at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method)
    at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269)
    at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93)
    at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86)
    - locked <307356c3> (a sun.nio.ch.Util$3)
    - locked <143004> (a java.util.Collections$UnmodifiableSet)
    - locked <3f90b948> (a sun.nio.ch.EPollSelectorImpl)
    at sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
    at org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:816)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-10" - Thread t@337
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-9" - Thread t@336
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-8" - Thread t@335
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)
   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-7" - Thread t@334
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-6" - Thread t@333
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
```

```
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-5" - Thread t@332
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-4" - Thread t@331
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-3" - Thread t@330
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-2" - Thread t@329
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"http-nio-auto-14-exec-1" - Thread t@328
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <1df7d9ee> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:108)
    at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:33)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None
```

```
"http-nio-auto-14-BlockPoller" - Thread t@327
   java.lang.Thread.State: RUNNABLE
    at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method)
    at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269)
    at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93)
    at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86)
    - locked <371851dd> (a sun.nio.ch.Util$3)
    - locked <16add32a> (a java.util.Collections$UnmodifiableSet)
    - locked <2ccd724e> (a sun.nio.ch.EPollSelectorImpl)
    at sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
    at
org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelect
or.java:331)

   Locked ownable synchronizers:
    - None

"Catalina-utility-2" - Thread t@326
   java.lang.Thread.State: TIMED_WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <f56e690> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos
(AbstractQueuedSynchronizer.java:2078)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1093)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"container-0" - Thread t@325
   java.lang.Thread.State: TIMED_WAITING
```

```
    at java.lang.Thread.sleep(Native Method)
    at org.apache.catalina.core.StandardServer.await(StandardServer.java:570)
    at
org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServ
er.java:197)

   Locked ownable synchronizers:
    - None

"Catalina-utility-1" - Thread t@324
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <f56e690> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1088)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"server" - Thread t@321
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <7a35d81e> (a
java.util.concurrent.CountDownLatch$Sync)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(Abst
ractQueuedSynchronizer.java:836)
    at
```

```
java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedInterruptib
ly(AbstractQueuedSynchronizer.java:997)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireSharedInterruptibly
(AbstractQueuedSynchronizer.java:1304)
    at java.util.concurrent.CountDownLatch.await(CountDownLatch.java:231)
    at
reactor.core.publisher.BlockingSingleSubscriber.blockingGet(BlockingSingleSubscr
iber.java:87)
    at reactor.core.publisher.Mono.block(Mono.java:1703)
    at
org.springframework.boot.web.embedded.netty.NettyWebServer$1.run(NettyWebServer.
java:180)

   Locked ownable synchronizers:
    - None

"HikariPool-1 housekeeper" - Thread t@302
   java.lang.Thread.State: TIMED_WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <27960f1> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos
(AbstractQueuedSynchronizer.java:2078)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1093)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"Keep-Alive-Timer" - Thread t@269
   java.lang.Thread.State: TIMED_WAITING
    at java.lang.Thread.sleep(Native Method)
    at sun.net.www.http.KeepAliveCache.run(KeepAliveCache.java:172)
    at java.lang.Thread.run(Thread.java:748)
   Locked ownable synchronizers:
    - None

"reactor-http-epoll-4" - Thread t@131
   java.lang.Thread.State: RUNNABLE
    at io.netty.channel.epoll.Native.epollWait(Native Method)
    at io.netty.channel.epoll.Native.epollWait(Native.java:177)
    at io.netty.channel.epoll.Native.epollWait(Native.java:170)
    at
io.netty.channel.epoll.EpollEventLoop.epollWaitNoTimerChange(EpollEventLoop.java
:290)
    at io.netty.channel.epoll.EpollEventLoop.run(EpollEventLoop.java:347)
    at
io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecut
or.java:989)
    at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
    at
io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.jav
a:30)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"reactor-http-epoll-3" - Thread t@130
   java.lang.Thread.State: RUNNABLE
    at io.netty.channel.epoll.Native.epollWait(Native Method)
    at io.netty.channel.epoll.Native.epollWait(Native.java:177)
    at io.netty.channel.epoll.Native.epollWait(Native.java:170)
    at
io.netty.channel.epoll.EpollEventLoop.epollWaitNoTimerChange(EpollEventLoop.java
:290)
    at io.netty.channel.epoll.EpollEventLoop.run(EpollEventLoop.java:347)
    at
io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecut
or.java:989)
    at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
    at
io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.jav
a:30)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"reactor-http-epoll-2" - Thread t@129
   java.lang.Thread.State: RUNNABLE
```

```
    at io.netty.channel.epoll.Native.epollWait(Native Method)
    at io.netty.channel.epoll.Native.epollWait(Native.java:177)
    at io.netty.channel.epoll.Native.epollWait(Native.java:170)
    at
io.netty.channel.epoll.EpollEventLoop.epollWaitNoTimerChange(EpollEventLoop.java
:290)
    at io.netty.channel.epoll.EpollEventLoop.run(EpollEventLoop.java:347)
    at
io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecut
or.java:989)
    at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
    at
io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.jav
a:30)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"reactor-http-epoll-1" - Thread t@128
   java.lang.Thread.State: RUNNABLE
    at io.netty.channel.epoll.Native.epollWait(Native Method)
    at io.netty.channel.epoll.Native.epollWait(Native.java:177)
    at io.netty.channel.epoll.Native.epollWait(Native.java:170)
    at
io.netty.channel.epoll.EpollEventLoop.epollWaitNoTimerChange(EpollEventLoop.java
:290)
    at io.netty.channel.epoll.EpollEventLoop.run(EpollEventLoop.java:347)
    at
io.netty.util.concurrent.SingleThreadEventExecutor$4.run(SingleThreadEventExecut
or.java:989)
    at io.netty.util.internal.ThreadExecutorMap$2.run(ThreadExecutorMap.java:74)
    at
io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.jav
a:30)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"boundedElastic-2" - Thread t@15
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <45613df2> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1081)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"boundedElastic-1" - Thread t@14
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <7d3b31a8> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1081)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"boundedElastic-evictor-1" - Thread t@13
   java.lang.Thread.State: TIMED_WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <291ae339> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos
(AbstractQueuedSynchronizer.java:2078)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:1093)
    at
java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(Scheduled
ThreadPoolExecutor.java:809)
    at
java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None

"/127.0.0.1:45976 to /127.0.0.1:42378 workers Thread 3" - Thread t@12
   java.lang.Thread.State: RUNNABLE
    at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method)
    at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269)
    at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93)
    at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86)
    - locked <60ad6003> (a sun.nio.ch.Util$3)
    - locked <2c6f4f49> (a java.util.Collections$UnmodifiableSet)
    - locked <87d8d1d> (a sun.nio.ch.EPollSelectorImpl)
    at sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
    at sun.nio.ch.SelectorImpl.select(SelectorImpl.java:101)
    at
org.gradle.internal.remote.internal.inet.SocketConnection$SocketInputStream.read
(SocketConnection.java:185)
    at com.esotericsoftware.kryo.io.Input.fill(Input.java:146)
    at com.esotericsoftware.kryo.io.Input.require(Input.java:178)
    at com.esotericsoftware.kryo.io.Input.readByte(Input.java:295)
    at
org.gradle.internal.serialize.kryo.KryoBackedDecoder.readByte(KryoBackedDecoder.
java:82)
    at
org.gradle.internal.remote.internal.hub.InterHubMessageSerializer$MessageReader.
read(InterHubMessageSerializer.java:64)
    at
org.gradle.internal.remote.internal.hub.InterHubMessageSerializer$MessageReader.
read(InterHubMessageSerializer.java:52)
```

```
    at
org.gradle.internal.remote.internal.inet.SocketConnection.receive(SocketConnecti
on.java:81)
    at
org.gradle.internal.remote.internal.hub.MessageHub$ConnectionReceive.run(Message
Hub.java:270)
    at
org.gradle.internal.concurrent.ExecutorPolicy$CatchAndRecordFailures.onExecute(E
xecutorPolicy.java:64)
    at
org.gradle.internal.concurrent.ManagedExecutorImpl$1.run(ManagedExecutorImpl.jav
a:48)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.gradle.internal.concurrent.ThreadFactoryImpl$ManagedThreadRunnable.run(Threa
dFactoryImpl.java:56)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - Locked <7ee955a8> (a java.util.concurrent.ThreadPoolExecutor$Worker)

"/127.0.0.1:45976 to /127.0.0.1:42378 workers Thread 2" - Thread t@11
   java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for <7600b400> (a
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at
java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abst
ractQueuedSynchronizer.java:2039)
    at
org.gradle.internal.remote.internal.hub.queue.EndPointQueue.take(EndPointQueue.j
ava:49)
    at
org.gradle.internal.remote.internal.hub.MessageHub$ConnectionDispatch.run(Messag
eHub.java:322)
    at
org.gradle.internal.concurrent.ExecutorPolicy$CatchAndRecordFailures.onExecute(E
xecutorPolicy.java:64)
    at
org.gradle.internal.concurrent.ManagedExecutorImpl$1.run(ManagedExecutorImpl.jav
a:48)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
```

```
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.gradle.internal.concurrent.ThreadFactoryImpl$ManagedThreadRunnable.run(Threa
dFactoryImpl.java:56)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - Locked <51e5fc98> (a java.util.concurrent.ThreadPoolExecutor$Worker)

"Test worker" - Thread t@10
   java.lang.Thread.State: RUNNABLE
    at sun.management.ThreadImpl.dumpThreads0(Native Method)
    at sun.management.ThreadImpl.dumpAllThreads(ThreadImpl.java:496)
    at sun.management.ThreadImpl.dumpAllThreads(ThreadImpl.java:484)
    at
org.springframework.boot.actuate.management.ThreadDumpEndpoint.getFormattedThrea
dDump(ThreadDumpEndpoint.java:51)
    at
org.springframework.boot.actuate.management.ThreadDumpEndpoint.textThreadDump(Th
readDumpEndpoint.java:47)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at
org.springframework.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:282)
    at
org.springframework.boot.actuate.endpoint.invoke.reflect.ReflectiveOperationInvo
ker.invoke(ReflectiveOperationInvoker.java:77)
    at
org.springframework.boot.actuate.endpoint.annotation.AbstractDiscoveredOperation
.invoke(AbstractDiscoveredOperation.java:60)
    at
org.springframework.boot.actuate.endpoint.web.servlet.AbstractWebMvcEndpointHand
lerMapping$ServletWebOperationAdapter.handle(AbstractWebMvcEndpointHandlerMappin
g.java:290)
    at
org.springframework.boot.actuate.endpoint.web.servlet.AbstractWebMvcEndpointHand
lerMapping$OperationHandler.handle(AbstractWebMvcEndpointHandlerMapping.java:373
)
    at sun.reflect.GeneratedMethodAccessor221.invoke(Unknown Source)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at
org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(Invocable
HandlerMethod.java:197)
    at
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(I
nvocableHandlerMethod.java:141)
    at
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMet
hod.invokeAndHandle(ServletInvocableHandlerMethod.java:106)
    at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapt
er.invokeHandlerMethod(RequestMappingHandlerAdapter.java:894)
    at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapt
er.handleInternal(RequestMappingHandlerAdapter.java:808)
    at
org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(A
bstractHandlerMethodAdapter.java:87)
    at
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.j
ava:1060)
    at
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.ja
va:962)
    at
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet
.java:1006)
    at
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:898
)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:645)
    at
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:8
83)
    at
org.springframework.test.web.servlet.TestDispatcherServlet.service(TestDispatche
rServlet.java:72)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:750)
    at
org.springframework.mock.web.MockFilterChain$ServletFilterProxy.doFilter(MockFil
terChain.java:167)
    at
org.springframework.mock.web.MockFilterChain.doFilter(MockFilterChain.java:134)
    at org.springframework.test.web.servlet.MockMvc.perform(MockMvc.java:183)
    at
org.springframework.boot.actuate.autoconfigure.endpoint.web.documentation.Thread
DumpEndpointDocumentationTests.textThreadDump(ThreadDumpEndpointDocumentationTes
ts.java:186)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at
org.junit.platform.commons.util.ReflectionUtils.invokeMethod(ReflectionUtils.jav
a:688)
    at
org.junit.jupiter.engine.execution.MethodInvocation.proceed(MethodInvocation.jav
a:60)
    at
org.junit.jupiter.engine.execution.InvocationInterceptorChain$ValidatingInvocati
on.proceed(InvocationInterceptorChain.java:131)
    at
org.junit.jupiter.engine.extension.TimeoutExtension.intercept(TimeoutExtension.j
ava:149)
    at
org.junit.jupiter.engine.extension.TimeoutExtension.interceptTestableMethod(Time
outExtension.java:140)
    at
org.junit.jupiter.engine.extension.TimeoutExtension.interceptTestMethod(TimeoutE
xtension.java:84)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor$$Lambda$129/5133968
32.apply(Unknown Source)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker$ReflectiveInterceptorCall.l
ambda$ofVoidMethod$0(ExecutableInvoker.java:115)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker$ReflectiveInterceptorCall$$
Lambda$130/564935064.apply(Unknown Source)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker.lambda$invoke$0(ExecutableI
nvoker.java:105)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker$$Lambda$252/873101565.apply
(Unknown Source)
    at
org.junit.jupiter.engine.execution.InvocationInterceptorChain$InterceptedInvocat
ion.proceed(InvocationInterceptorChain.java:106)
    at
org.junit.jupiter.engine.execution.InvocationInterceptorChain.proceed(Invocation
InterceptorChain.java:64)
    at
org.junit.jupiter.engine.execution.InvocationInterceptorChain.chainAndInvoke(Inv
ocationInterceptorChain.java:45)
    at
```

```
org.junit.jupiter.engine.execution.InvocationInterceptorChain.invoke(InvocationI
nterceptorChain.java:37)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker.invoke(ExecutableInvoker.ja
va:104)
    at
org.junit.jupiter.engine.execution.ExecutableInvoker.invoke(ExecutableInvoker.ja
va:98)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.lambda$invokeTestMe
thod$6(TestMethodTestDescriptor.java:210)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor$$Lambda$291/1938180
201.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.invokeTestMethod(Te
stMethodTestDescriptor.java:206)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.execute(TestMethodT
estDescriptor.java:131)
    at
org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.execute(TestMethodT
estDescriptor.java:65)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$5(NodeTestTask.java:139)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$199/76523088
9.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$7(NodeTestTask.java:129)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$198/23761766
1.invoke(Unknown Source)
    at org.junit.platform.engine.support.hierarchical.Node.around(Node.java:137)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$8(NodeTestTask.java:127)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$197/15196372
66.execute(Unknown Source)
    at
```

```
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.executeRecursively(N
odeTestTask.java:126)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.execute(NodeTestTask
.java:84)
    at
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecuto
rService$$Lambda$203/2050582666.accept(Unknown Source)
    at java.util.ArrayList.forEach(ArrayList.java:1259)
    at
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecuto
rService.invokeAll(SameThreadHierarchicalTestExecutorService.java:38)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$5(NodeTestTask.java:143)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$199/76523088
9.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$7(NodeTestTask.java:129)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$198/23761766
1.invoke(Unknown Source)
    at org.junit.platform.engine.support.hierarchical.Node.around(Node.java:137)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$8(NodeTestTask.java:127)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$197/15196372
66.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.executeRecursively(N
odeTestTask.java:126)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.execute(NodeTestTask
.java:84)
    at
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecuto
rService$$Lambda$203/2050582666.accept(Unknown Source)
```

```
    at java.util.ArrayList.forEach(ArrayList.java:1259)
    at
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecuto
rService.invokeAll(SameThreadHierarchicalTestExecutorService.java:38)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$5(NodeTestTask.java:143)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$199/76523088
9.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$7(NodeTestTask.java:129)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$198/23761766
1.invoke(Unknown Source)
    at org.junit.platform.engine.support.hierarchical.Node.around(Node.java:137)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.lambda$executeRecurs
ively$8(NodeTestTask.java:127)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask$$Lambda$197/15196372
66.execute(Unknown Source)
    at
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(Throwa
bleCollector.java:73)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.executeRecursively(N
odeTestTask.java:126)
    at
org.junit.platform.engine.support.hierarchical.NodeTestTask.execute(NodeTestTask
.java:84)
    at
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecuto
rService.submit(SameThreadHierarchicalTestExecutorService.java:32)
    at
org.junit.platform.engine.support.hierarchical.HierarchicalTestExecutor.execute(
HierarchicalTestExecutor.java:57)
    at
org.junit.platform.engine.support.hierarchical.HierarchicalTestEngine.execute(Hi
erarchicalTestEngine.java:51)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator.execute(EngineExecu
tionOrchestrator.java:108)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator.execute(EngineExecu
tionOrchestrator.java:88)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator.lambda$execute$0(En
gineExecutionOrchestrator.java:54)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator$$Lambda$158/2657904
45.accept(Unknown Source)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator.withInterceptedStre
ams(EngineExecutionOrchestrator.java:67)
    at
org.junit.platform.launcher.core.EngineExecutionOrchestrator.execute(EngineExecu
tionOrchestrator.java:52)
    at
org.junit.platform.launcher.core.DefaultLauncher.execute(DefaultLauncher.java:96
)
    at
org.junit.platform.launcher.core.DefaultLauncher.execute(DefaultLauncher.java:75
)
    at
org.gradle.api.internal.tasks.testing.junitplatform.JUnitPlatformTestClassProces
sor$CollectAllTestClassesExecutor.processAllTestClasses(JUnitPlatformTestClassPr
ocessor.java:99)
    at
org.gradle.api.internal.tasks.testing.junitplatform.JUnitPlatformTestClassProces
sor$CollectAllTestClassesExecutor.access$000(JUnitPlatformTestClassProcessor.jav
a:79)
    at
org.gradle.api.internal.tasks.testing.junitplatform.JUnitPlatformTestClassProces
sor.stop(JUnitPlatformTestClassProcessor.java:75)
    at
org.gradle.api.internal.tasks.testing.SuiteTestClassProcessor.stop(SuiteTestClas
sProcessor.java:61)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at
org.gradle.internal.dispatch.ReflectionDispatch.dispatch(ReflectionDispatch.java
:36)
    at
org.gradle.internal.dispatch.ReflectionDispatch.dispatch(ReflectionDispatch.java
:24)
    at
org.gradle.internal.dispatch.ContextClassLoaderDispatch.dispatch(ContextClassLoa
derDispatch.java:33)
```

```
    at
org.gradle.internal.dispatch.ProxyDispatchAdapter$DispatchingInvocationHandler.i
nvoke(ProxyDispatchAdapter.java:94)
    at com.sun.proxy.$Proxy2.stop(Unknown Source)
    at
org.gradle.api.internal.tasks.testing.worker.TestWorker.stop(TestWorker.java:133
)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.jav
a:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at
org.gradle.internal.dispatch.ReflectionDispatch.dispatch(ReflectionDispatch.java
:36)
    at
org.gradle.internal.dispatch.ReflectionDispatch.dispatch(ReflectionDispatch.java
:24)
    at
org.gradle.internal.remote.internal.hub.MessageHubBackedObjectConnection$Dispatc
hWrapper.dispatch(MessageHubBackedObjectConnection.java:182)
    at
org.gradle.internal.remote.internal.hub.MessageHubBackedObjectConnection$Dispatc
hWrapper.dispatch(MessageHubBackedObjectConnection.java:164)
    at
org.gradle.internal.remote.internal.hub.MessageHub$Handler.run(MessageHub.java:4
14)
    at
org.gradle.internal.concurrent.ExecutorPolicy$CatchAndRecordFailures.onExecute(E
xecutorPolicy.java:64)
    at
org.gradle.internal.concurrent.ManagedExecutorImpl$1.run(ManagedExecutorImpl.jav
a:48)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at
org.gradle.internal.concurrent.ThreadFactoryImpl$ManagedThreadRunnable.run(Threa
dFactoryImpl.java:56)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - Locked <4fe767f3> (a java.util.concurrent.ThreadPoolExecutor$Worker)

"Signal Dispatcher" - Thread t@4
```

```
   java.lang.Thread.State: RUNNABLE

   Locked ownable synchronizers:
    - None

"Finalizer" - Thread t@3
   java.lang.Thread.State: WAITING
    at java.lang.Object.wait(Native Method)
    - waiting on <7da67fbb> (a java.lang.ref.ReferenceQueue$Lock)
    at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
    at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
    at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)

   Locked ownable synchronizers:
    - None

"Reference Handler" - Thread t@2
   java.lang.Thread.State: WAITING
    at java.lang.Object.wait(Native Method)
    - waiting on <2be57241> (a java.lang.ref.Reference$Lock)
    at java.lang.Object.wait(Object.java:502)
    at java.lang.ref.Reference.tryHandlePending(Reference.java:191)
    at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)

   Locked ownable synchronizers:
    - None
```