How to show the contents of log file in browser screen using Spring Actuator?

Asked 5 years, 8 months ago Modified 1 year, 8 months ago Viewed 10k times

2

I'm trying to show the contents of a logger file in the browser screen , so that when the application is running in production in external server, i don't need to login every time into the server to fetch the logs. I'm trying to achieve this using Spring Boot Actuator. I have configured the log file path and log info level in my properties file, and logs are being written in that file, but how to stream the contents of the file in browser window. below is my properties file contents

```
management.security.enabled=false
endpoints.env.enabled=false
endpoints.configprops.enabled=false
endpoints.autoconfig.enabled=false
endpoints.beans.enabled=false
endpoints.dump.enabled=true
endpoints.heapdump.enabled=true
logging.level.root=info
logging.file=target/app.log
```

Thanks for the help in advance !!!!

java spring-boot spring-mvc spring-logback

Share Edit Follow

asked Mar 7, 2019 at 7:49
Pradeep Anand
155 ●2 ●4 ●15

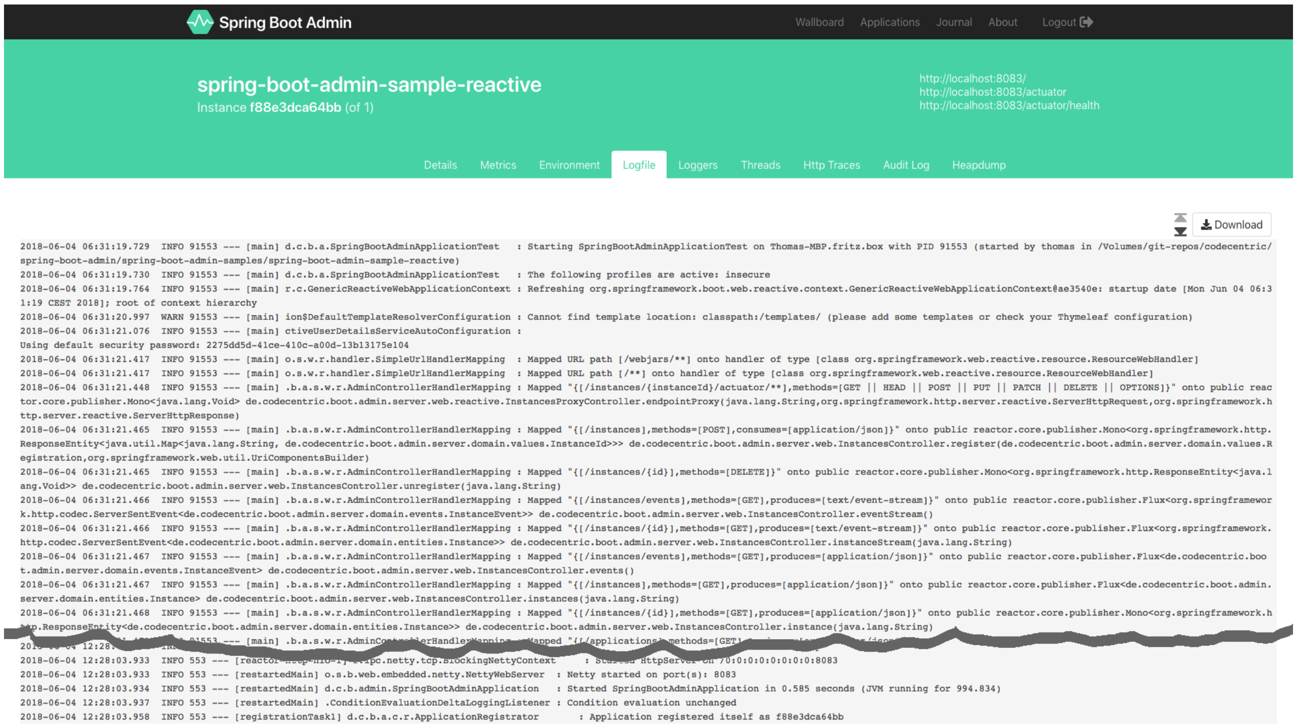
3 Answers

Sorted by: Highest score (default)

1

You can use Spring Boot Admin: https://github.com/codecentric/spring-boot-admin

The logs appear like this:



```
2018-06-04 06:31:19.729  INFO 91553 --- [main] d.c.b.a.SpringBootAdminApplicationTest   : Starting SpringBootAdminApplicationTest on Thomas-MBP.fritz.box with PID 91553 (started by thomas in /Volumes/git-repos/codecentric/
spring-boot-admin/spring-boot-admin-samples/spring-boot-admin-sample-reactive)
2018-06-04 06:31:19.730  INFO 91553 --- [main] d.c.b.a.SpringBootAdminApplicationTest   : The following profiles are active: insecure
2018-06-04 06:31:19.764  INFO 91553 --- [main] r.c.GenericReactiveWebApplicationContext : Refreshing org.springframework.boot.web.reactive.context.GenericReactiveWebApplicationContext@ae3540e: startup date [Mon Jun 04 06:3
1:19 CEST 2018]; root of context hierarchy
2018-06-04 06:31:20.997  WARN 91553 --- [main] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/ (please add some templates or check your Thymeleaf configuration)
2018-06-04 06:31:21.076  INFO 91553 --- [main] ctiveUserDetailsServiceAutoConfiguration :
Using default security password: 2275dd5d-41ce-410c-a00d-13b13175e104
2018-06-04 06:31:21.417  INFO 91553 --- [main] o.s.w.r.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.reactive.resource.ResourceWebHandler]
2018-06-04 06:31:21.417  INFO 91553 --- [main] o.s.w.r.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.springframework.web.reactive.resource.ResourceWebHandler]
2018-06-04 06:31:21.448  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/{instanceId}/actuator/**],methods=[GET || HEAD || POST || PUT || PATCH || DELETE || OPTIONS]}" onto public reac
tor.core.publisher.Mono<java.lang.Void> de.codecentric.boot.admin.server.web.reactive.InstancesProxyController.endpointProxy(java.lang.String,org.springframework.http.server.reactive.ServerHttpRequest,org.springframework.h
ttp.server.reactive.ServerHttpResponse)
2018-06-04 06:31:21.465  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances],methods=[POST],consumes=[application/json]}" onto public reactor.core.publisher.Mono<org.springframework.http.
ResponseEntity<java.util.Map<java.lang.String, de.codecentric.boot.admin.server.domain.values.InstanceId>>> de.codecentric.boot.admin.server.web.InstancesController.register(de.codecentric.boot.admin.server.domain.values.R
egistration,org.springframework.web.util.UriComponentsBuilder)
2018-06-04 06:31:21.465  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/{id}],methods=[DELETE]}" onto public reactor.core.publisher.Mono<org.springframework.http.ResponseEntity<java.l
ang.Void>> de.codecentric.boot.admin.server.web.InstancesController.unregister(java.lang.String)
2018-06-04 06:31:21.466  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/events],methods=[GET],produces=[text/event-stream]}" onto public reactor.core.publisher.Flux<org.springframewor
k.http.codec.ServerSentEvent<de.codecentric.boot.admin.server.domain.events.InstanceEvent>> de.codecentric.boot.admin.server.web.InstancesController.eventStream()
2018-06-04 06:31:21.466  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/{id}],methods=[GET],produces=[text/event-stream]}" onto public reactor.core.publisher.Flux<org.springframework.
http.codec.ServerSentEvent<de.codecentric.boot.admin.server.domain.entities.Instance>> de.codecentric.boot.admin.server.web.InstancesController.instanceStream(java.lang.String)
2018-06-04 06:31:21.467  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/events],methods=[GET],produces=[application/json]}" onto public reactor.core.publisher.Flux<de.codecentric.boo
t.admin.server.domain.events.InstanceEvent> de.codecentric.boot.admin.server.web.InstancesController.events()
2018-06-04 06:31:21.467  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances],methods=[GET],produces=[application/json]}" onto public reactor.core.publisher.Flux<de.codecentric.boot.admin.
server.domain.entities.Instance> de.codecentric.boot.admin.server.web.InstancesController.instances(java.lang.String)
2018-06-04 06:31:21.468  INFO 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/instances/{id}],methods=[GET],produces=[application/json]}" onto public reactor.core.publisher.Mono<org.springframework.h
ttp.ResponseEntity<de.codecentric.boot.admin.server.domain.entities.Instance>> de.codecentric.boot.admin.server.web.InstancesController.instance(java.lang.String)
[illegible] 91553 --- [main] .b.a.s.w.r.AdminControllerHandlerMapping : Mapped "{[/applications],methods=[GET] [illegible]
2018-06-04 12:28:[illegible]
2018-06-04 12:28:03.933  INFO 553 --- [reactor-http-nio-1] r.ipc.netty.tcp.BlockingNettyContext     : Started HttpServer on /0:0:0:0:0:0:0:0:8083
2018-06-04 12:28:03.933  INFO 553 --- [restartedMain] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port(s): 8083
2018-06-04 12:28:03.934  INFO 553 --- [restartedMain] d.c.b.admin.SpringBootAdminApplication   : Started SpringBootAdminApplication in 0.585 seconds (JVM running for 994.834)
2018-06-04 12:28:03.937  INFO 553 --- [restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
2018-06-04 12:28:03.958  INFO 553 --- [registrationTask1] d.c.b.a.c.r.ApplicationRegistrator       : Application registered itself as f88e3dca64bb
```

You can use https://start.spring.io/ to include de Admin Client and Server in your project. Check the tutorial here: http://codecentric.github.io/spring-boot-admin/current/#getting-started

Share Edit Follow

answered Mar 7, 2019 at 7:55

voliveira89
1,274 ● 2 ● 12 ● 23

1

By default, the following Spring Boot Actuator endpoints are enabled (JMX/WEB):

https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-features.html#production-ready-endpoints-exposing-endpoints

To enable specific endpoints write the following in Spring Boot `application.properties` file:

```
management.endpoints.web.exposure.include = info, health, logfile
```

or to disable write:

```
management.endpoints.web.exposure.exclude = env,beans
```

Share Edit Follow

answered Jan 17, 2021 at 13:24

Šime Tokić
710 ● 1 ● 9 ● 23

Logs in the browser:

0

One way to expose the logs from a Spring Boot application in the browser is by using the built-in Actuator endpoints.

Actuator is a tool that provides production-ready features to help you monitor and manage your Spring Boot application. It includes a variety of endpoints, including one for viewing the application logs.

To enable the Actuator endpoints, you need to add the spring-boot-starter-actuator dependency to your project's pom.xml file:

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

Once the dependency is added, you can access the log endpoint at `http://localhost:8080/actuator/logfile`.

To expose the logs in the browser, you can create a custom endpoint that reads the log file and returns its contents. Here's an example:

```
@RestController
public class LogController {

    @GetMapping("/logs")
    public ResponseEntity<String> getLogs() throws IOException {
        Path logFile = Paths.get(System.getProperty("logging.file.name"));
        String logs = new String(Files.readAllBytes(logFile));
        return ResponseEntity.ok(logs);
    }
}
```

Make sure the application is configured to write logs into a file, by adding the following to the `application.properties` or `application.yml` file:

```
logging.file.name=mylog.log
logging.level.root=INFO
```

This code creates a new endpoint at `http://localhost:8080/logs` that returns the contents of the log file as a string.

Note that **this approach may not be suitable for production environments** as it exposes potentially sensitive information. It is recommended to restrict access to the log endpoint using appropriate security measures.

Share Edit Follow

answered Mar 5, 2023 at 2:43

Nacho

119 ● 1 ● 3