Finite State machine

Finite models are essential in testing for simplifying complex systems, allowing exhaustive testing, and facilitating automated test case generation. They offer a clear way to represent systems as finite state machines, making it possible to explore all possible states and transitions. This approach is critical for ensuring systems behave as expected under every scenario, especially in critical applications where failure is unacceptable.

Key benefits include:

- Simplification: Reducing complex systems to manageable models.
- Exhaustive Testing: Enabling complete coverage of all possible states and transitions.
- Automated Testing: Supporting the generation of test cases and regression testing.
- Formal Verification: Allowing mathematical proofs of system correctness through model checking and other formal methods.

Finite models are widely used in software and hardware testing to improve test coverage, find potential issues, and ensure system reliability. However, challenges such as scalability and the accuracy of the models must be managed to ensure effective testing outcomes.

Srping boot actuator

Spring boot actuator is one of the most important component from spring boot.provides a set of ready-to-use features to help you monitor and manage your application. These features allow you to inspect the internal state of your application, such as health status, metrics, environment information, and more. This is especially useful for microservices architectures in production environments.

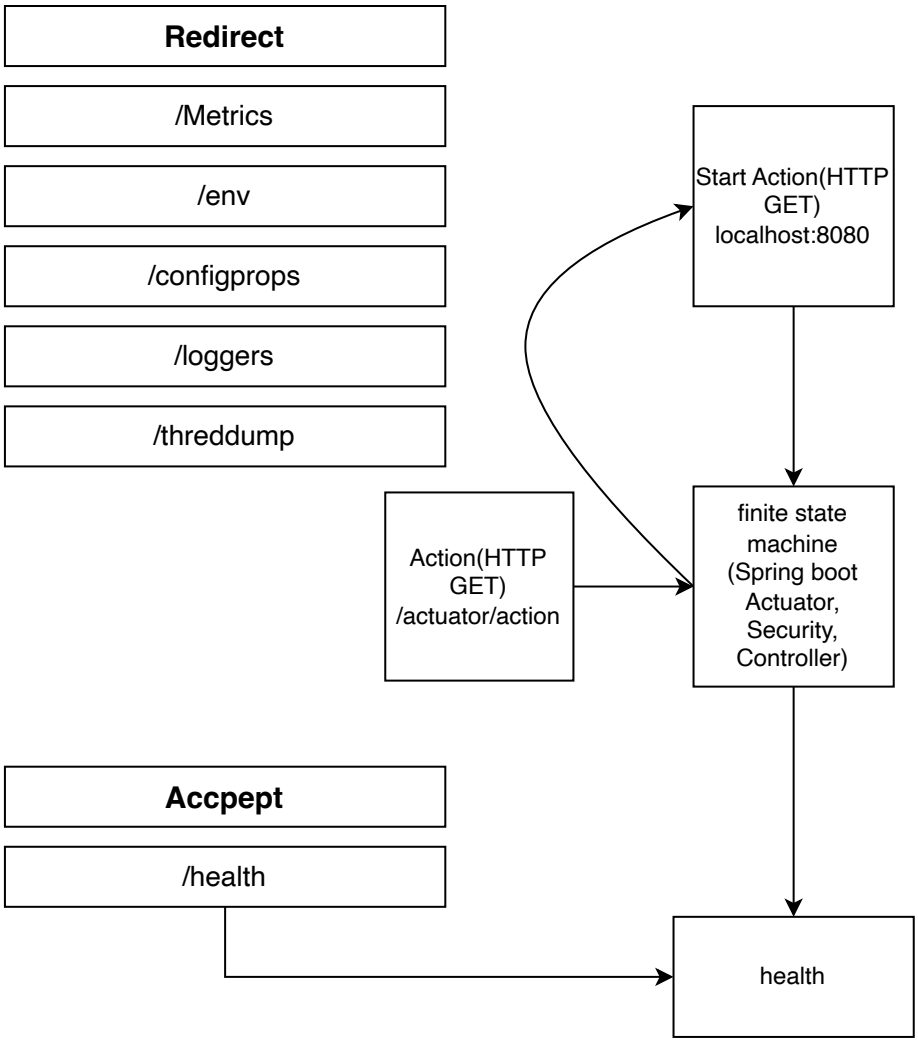
There are some endpoints from actuator:

- `/actuator` : The root endpoint that lists all available actuator endpoints.
- `/health` : Summarizes the health status of your application.
- `/heapdump` : Triggers a dump of the Java heap.
- `/httptrace` : Shows HTTP trace information (by default, the last 100 HTTP request-response exchanges).
- `/info` : Displays arbitrary application info.
- `/loggers` : Enables viewing and modifying the logging level of application loggers.
- `/metrics` : Shows ‘metrics’ information for the current application.
- `/shutdown` : Lets you gracefully shut down the application (disabled by default).

- `/env` : Exposes properties from Spring's ConfigurableEnvironment.

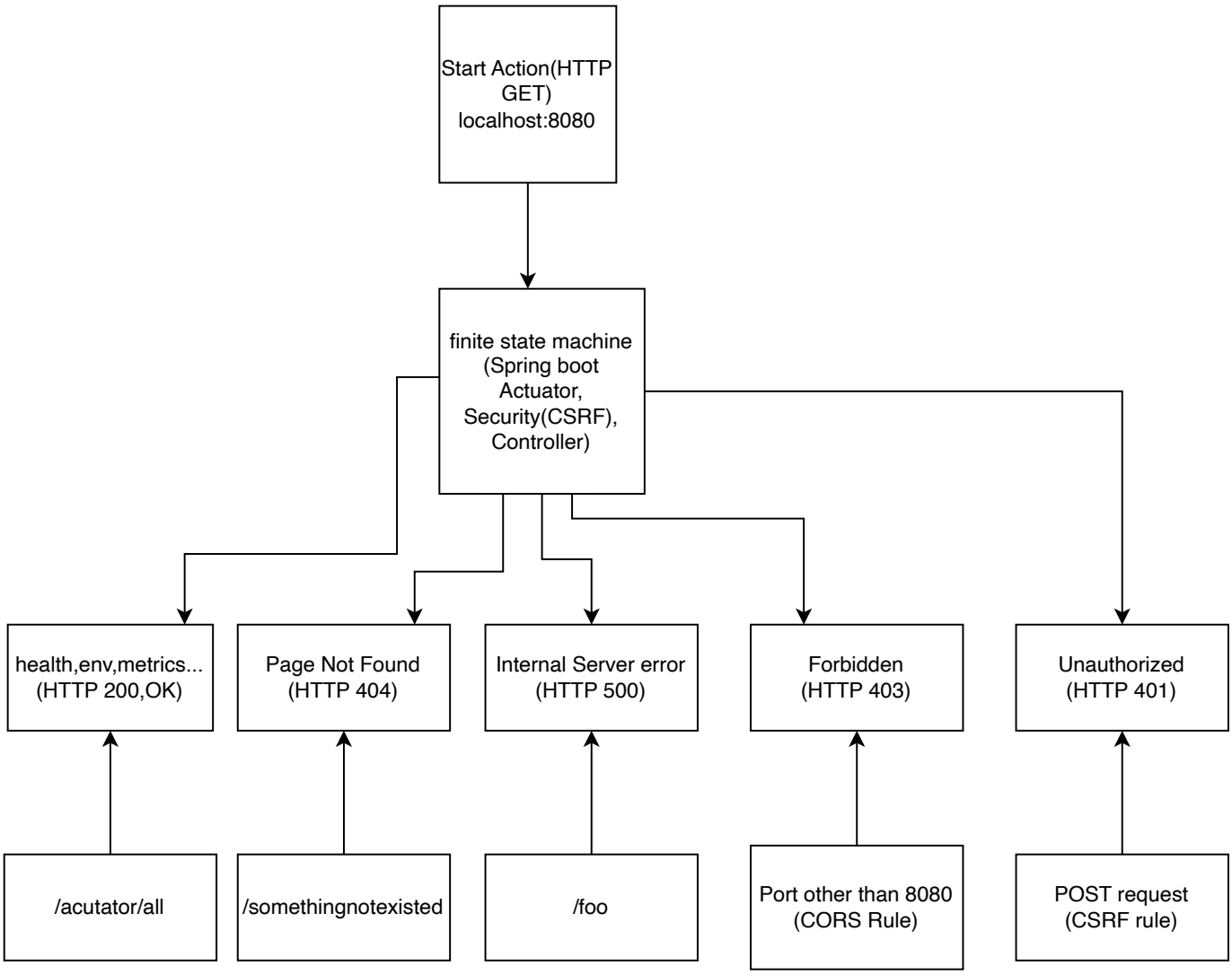
This is the function what actuator offers; I won't explain the detailed functions here. The spring boot basic policy is: **All the function except /health will be treated as sensitive data which will not open to the public until the auth passed.**

The finite state machine without verification will be look like this:



All the actuator function except health will be received a http 302 redirect response. The health will be received http ok and show the data query.

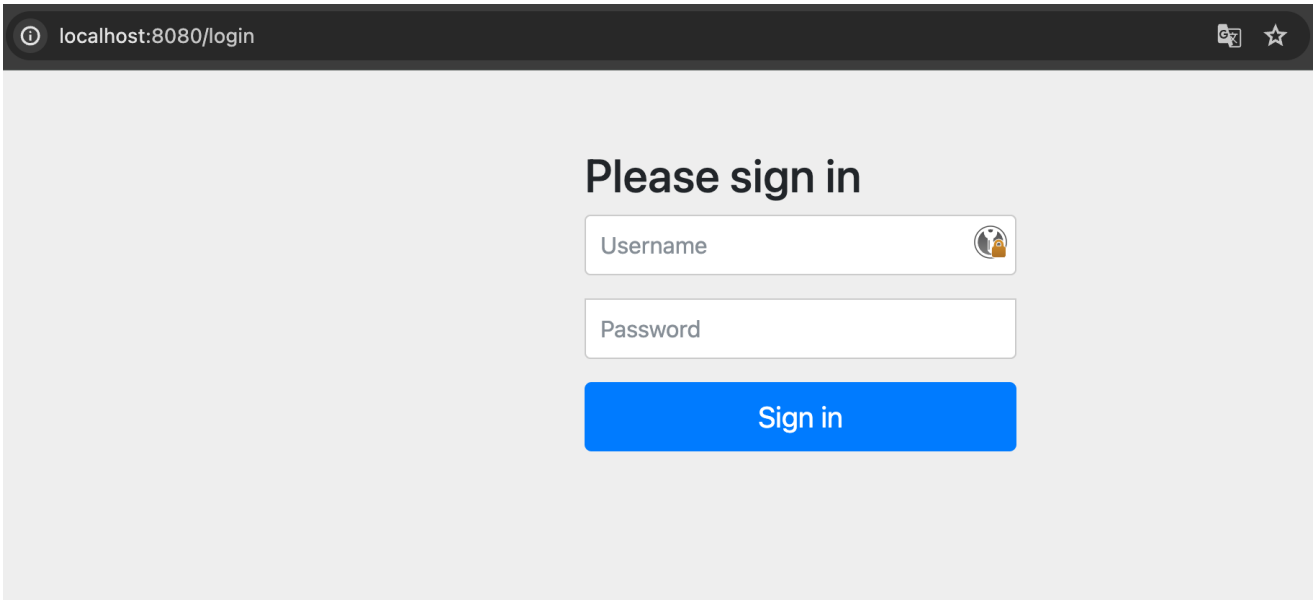
Let's see how the finite state machine change when verification completed.



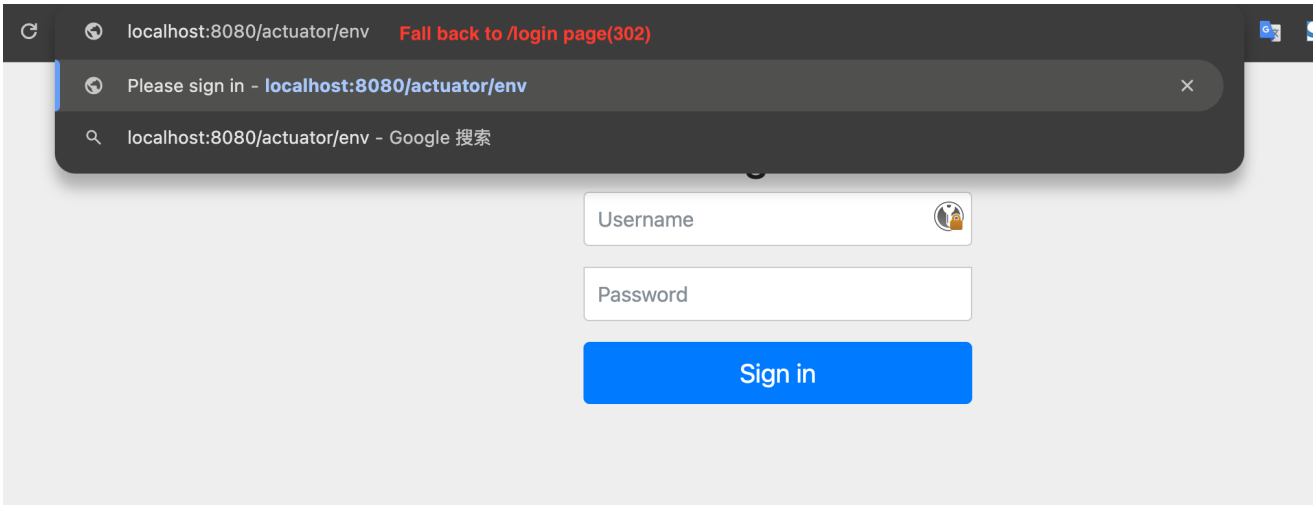
Now, the server will make more response, not just including all the actuator services, but also the exception handling. like `/foo` will response http500. http 403 will pop up when make a GET request in prohibited port when CORS enable.

Below shows a bunch of screenshots about how it actually reacts:

first the login page

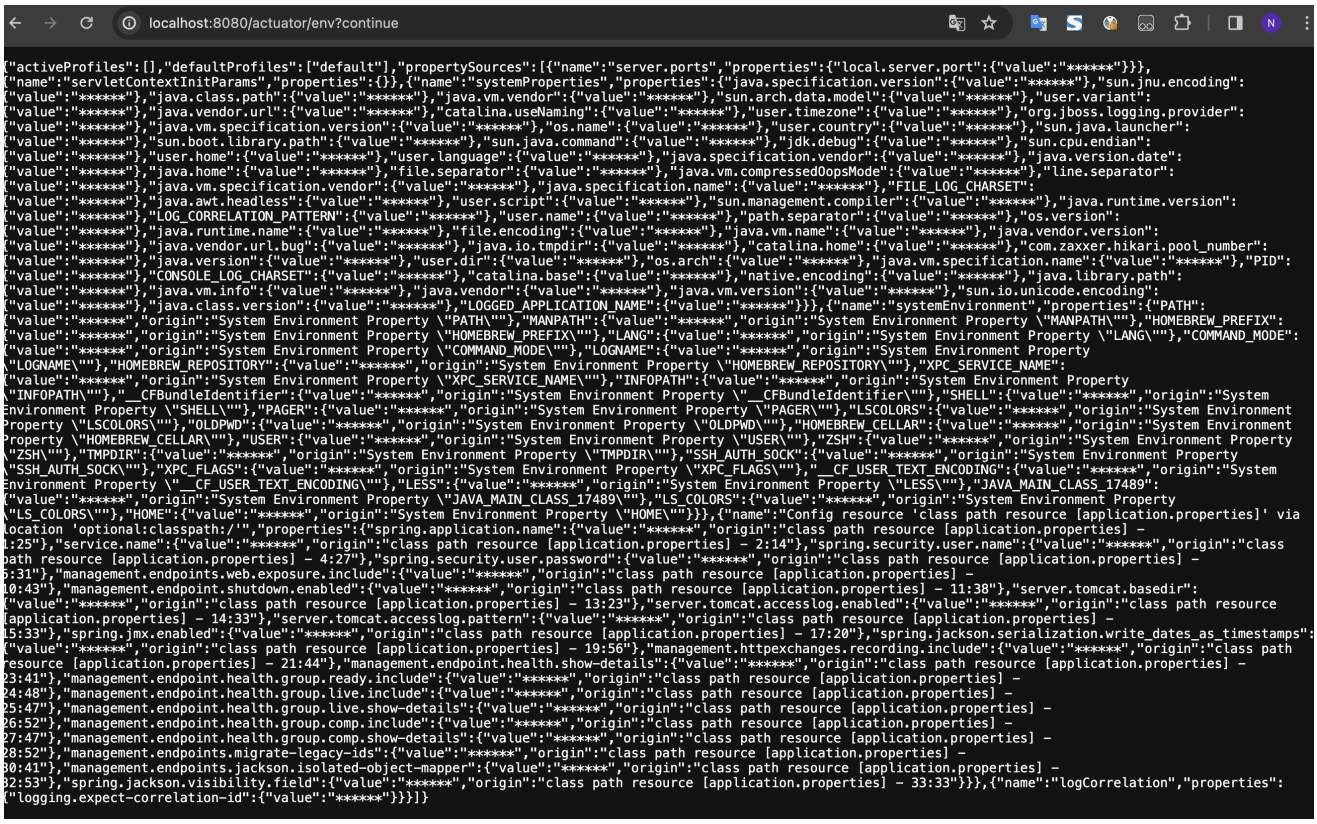


try to enter actuator/env and excute, page automatically fall back to login page



try use `actuator/health` get the new return

`/actuator/env` now functional after entering correct username and password




```
{"activeProfiles":[],"defaultProfiles":["default"],"propertySources":[{"name":"server.ports","properties":{"local.server.port":{"value":"******"}}},
{"name":"servletContextInitParams","properties":{}},{"name":"systemProperties","properties":{"java.specification.version":{"value":"******"},"sun.jnu.encoding":
{"value":"******"},"java.class.path":{"value":"******"},"java.vm.vendor":{"value":"******"},"sun.arch.data.model":{"value":"******"},"user.variant":
{"value":"******"},"java.vendor.url":{"value":"******"},"catalina.useNaming":{"value":"******"},"user.timezone":{"value":"******"},"org.jboss.logging.provider":
{"value":"******"},"java.vm.specification.version":{"value":"******"},"os.name":{"value":"******"},"user.country":{"value":"******"},"sun.java.launcher":
{"value":"******"},"sun.boot.library.path":{"value":"******"},"sun.java.command":{"value":"******"},"jdk.debug":{"value":"******"},"sun.cpu.endian":
{"value":"******"},"user.home":{"value":"******"},"user.language":{"value":"******"},"java.specification.vendor":{"value":"******"},"java.version.date":
{"value":"******"},"java.home":{"value":"******"},"file.separator":{"value":"******"},"java.vm.compressedOopsMode":{"value":"******"},"line.separator":
{"value":"******"},"java.vm.specification.vendor":{"value":"******"},"java.specification.name":{"value":"******"},"FILE_LOG_CHARSET":
{"value":"******"},"java.awt.headless":{"value":"******"},"user.script":{"value":"******"},"sun.management.compiler":{"value":"******"},"java.runtime.version":
{"value":"******"},"LOG_CORRELATION_PATTERN":{"value":"******"},"user.name":{"value":"******"},"path.separator":{"value":"******"},"os.version":
{"value":"******"},"java.runtime.name":{"value":"******"},"file.encoding":{"value":"******"},"java.vm.name":{"value":"******"},"java.vendor.version":
{"value":"******"},"java.vendor.url.bug":{"value":"******"},"java.io.tmpdir":{"value":"******"},"catalina.home":{"value":"******"},"com.zaxxer.hikari.pool_number":
{"value":"******"},"java.version":{"value":"******"},"user.dir":{"value":"******"},"os.arch":{"value":"******"},"java.vm.specification.name":{"value":"******"},"PID":
{"value":"******"},"CONSOLE_LOG_CHARSET":{"value":"******"},"catalina.base":{"value":"******"},"native.encoding":{"value":"******"},"java.library.path":
{"value":"******"},"java.vm.info":{"value":"******"},"java.vendor":{"value":"******"},"java.vm.version":{"value":"******"},"sun.io.unicode.encoding":
{"value":"******"},"java.class.version":{"value":"******"},"LOGGED_APPLICATION_NAME":{"value":"******"}}},{"name":"systemEnvironment","properties":{"PATH":
{"value":"******","origin":"System Environment Property \"PATH\""},"MANPATH":{"value":"******","origin":"System Environment Property \"MANPATH\""},"HOMEBREW_PREFIX":
{"value":"******","origin":"System Environment Property \"HOMEBREW_PREFIX\""},"LANG":{"value":"******","origin":"System Environment Property \"LANG\""},"COMMAND_MODE":
{"value":"******","origin":"System Environment Property \"COMMAND_MODE\""},"LOGNAME":{"value":"******","origin":"System Environment Property
\"LOGNAME\""},"HOMEBREW_REPOSITORY":{"value":"******","origin":"System Environment Property \"HOMEBREW_REPOSITORY\""},"XPC_SERVICE_NAME":
{"value":"******","origin":"System Environment Property \"XPC_SERVICE_NAME\""},"INFOPATH":{"value":"******","origin":"System Environment Property
\"INFOPATH\""},"__CFBundleIdentifier":{"value":"******","origin":"System Environment Property \"__CFBundleIdentifier\""},"SHELL":{"value":"******","origin":"System
Environment Property \"SHELL\""},"PAGER":{"value":"******","origin":"System Environment Property \"PAGER\""},"LSCOLORS":{"value":"******","origin":"System Environment
Property \"LSCOLORS\""},"OLDPWD":{"value":"******","origin":"System Environment Property \"OLDPWD\""},"HOMEBREW_CELLAR":{"value":"******","origin":"System Environment
Property \"HOMEBREW_CELLAR\""},"USER":{"value":"******","origin":"System Environment Property \"USER\""},"ZSH":{"value":"******","origin":"System Environment Property
\"ZSH\""},"TMPDIR":{"value":"******","origin":"System Environment Property \"TMPDIR\""},"SSH_AUTH_SOCK":{"value":"******","origin":"System Environment Property
\"SSH_AUTH_SOCK\""},"XPC_FLAGS":{"value":"******","origin":"System Environment Property \"XPC_FLAGS\""},"__CF_USER_TEXT_ENCODING":{"value":"******","origin":"System
Environment Property \"__CF_USER_TEXT_ENCODING\""},"LESS":{"value":"******","origin":"System Environment Property \"LESS\""},"JAVA_MAIN_CLASS_17489":
{"value":"******","origin":"System Environment Property \"JAVA_MAIN_CLASS_17489\""},"LS_COLORS":{"value":"******","origin":"System Environment Property
\"LS_COLORS\""},"HOME":{"value":"******","origin":"System Environment Property \"HOME\""}}},{"name":"Config resource 'class path resource [application.properties]' via
location 'optional:classpath:/'","properties":{"spring.application.name":{"value":"******","origin":"class path resource [application.properties] -
1:25"},"service.name":{"value":"******","origin":"class path resource [application.properties] - 2:14"},"spring.security.user.name":{"value":"******","origin":"class
path resource [application.properties] - 4:27"},"spring.security.user.password":{"value":"******","origin":"class path resource [application.properties] -
5:31"},"management.endpoints.web.exposure.include":{"value":"******","origin":"class path resource [application.properties] -
10:43"},"management.endpoint.shutdown.enabled":{"value":"******","origin":"class path resource [application.properties] - 11:38"},"server.tomcat.basedir":
{"value":"******","origin":"class path resource [application.properties] - 13:23"},"server.tomcat.accesslog.enabled":{"value":"******","origin":"class path resource
[application.properties] - 14:33"},"server.tomcat.accesslog.pattern":{"value":"******","origin":"class path resource [application.properties] -
15:33"},"spring.jmx.enabled":{"value":"******","origin":"class path resource [application.properties] - 17:20"},"spring.jackson.serialization.write_dates_as_timestamps":
{"value":"******","origin":"class path resource [application.properties] - 19:56"},"management.httpexchanges.recording.include":{"value":"******","origin":"class path
resource [application.properties] - 21:44"},"management.endpoint.health.show-details":{"value":"******","origin":"class path resource [application.properties] -
23:41"},"management.endpoint.health.group.ready.include":{"value":"******","origin":"class path resource [application.properties] -
24:48"},"management.endpoint.health.group.live.include":{"value":"******","origin":"class path resource [application.properties] -
25:47"},"management.endpoint.health.group.live.show-details":{"value":"******","origin":"class path resource [application.properties] -
26:52"},"management.endpoint.health.group.comp.include":{"value":"******","origin":"class path resource [application.properties] -
27:47"},"management.endpoint.health.group.comp.show-details":{"value":"******","origin":"class path resource [application.properties] -
28:52"},"management.endpoints.migrate-legacy-ids":{"value":"******","origin":"class path resource [application.properties] -
30:41"},"management.endpoints.jackson.isolated-object-mapper":{"value":"******","origin":"class path resource [application.properties] -
32:53"},"spring.jackson.visibility.field":{"value":"******","origin":"class path resource [application.properties] - 33:33"}}},{"name":"logCorrelation","properties":
{"logging.expect-correlation-id":{"value":"******"}}}]}
```

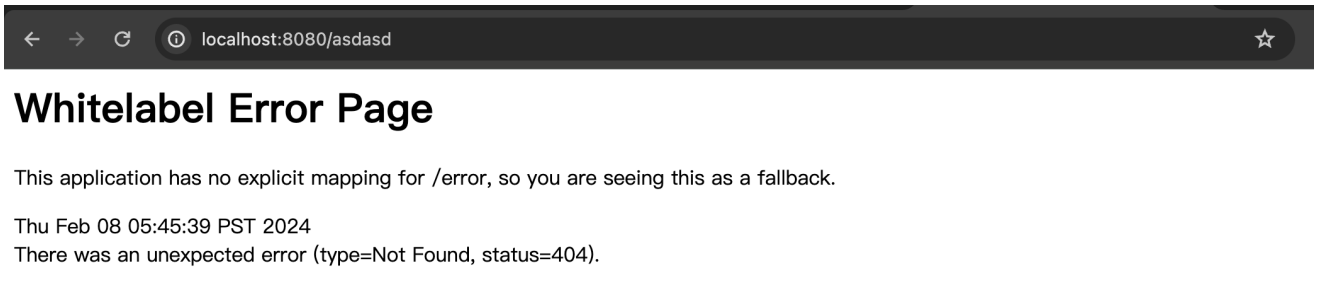

try exception page, get http 500

Whitelabel Error Page

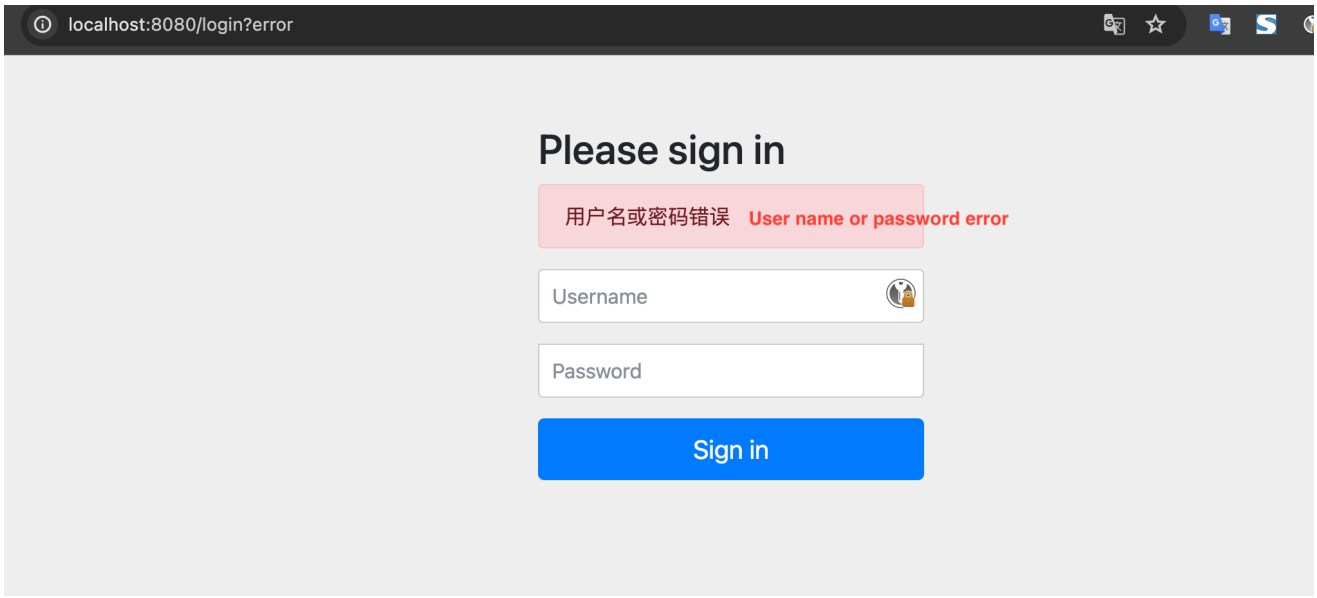
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Thu Feb 08 05:44:56 PST 2024
There was an unexpected error (type=Internal Server Error, status=500).

try to get some page not existed, get http 404



login fail can also be a simple state machine, but it's code implementation based on spring security, hence not within the scope of this discussion



All new test cases append to ▶ `SampleActuatorApplicationTests`