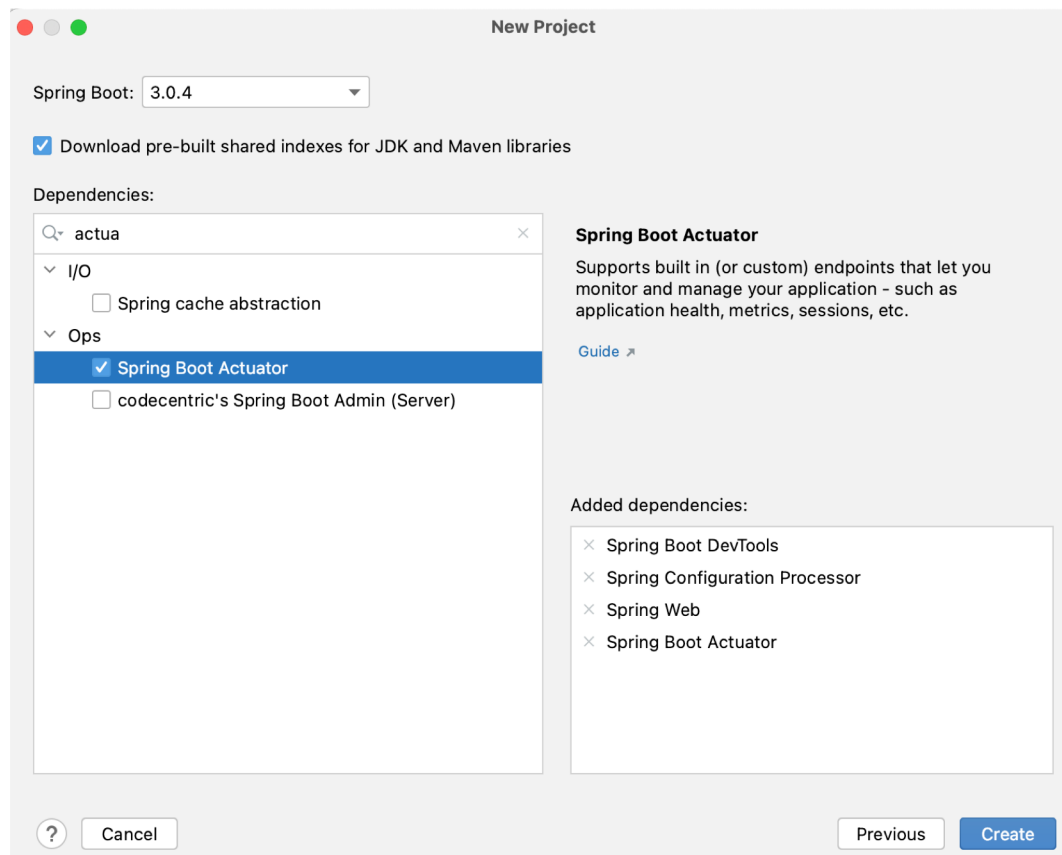


Actuator

Actuator

❖ Actuator

✓ 프로젝트 생성



EndPoint

❖ EndPoint

- ✓ 액추에이터의 EndPoint는 애플리케이션의 모니터링을 사용하는 경로
- ✓ 스프링 부트에는 여러 내장 EndPoint가 포함돼 있으며 커스텀 EndPoint를 추가할 수도 있음
- ✓ 액추에이터를 추가하면 기본적으로 EndPoint URL로 /actuator가 추가되며 이 뒤에 경로를 추가해 상세 내역에 접근
- ✓ actuator 경로가 아닌 다른 경로를 사용하고 싶다면 application.properties 파일에 작성
`management.endpoints.web.base-path=/custom-path`

EndPoint

❖ EndPoint

✓ 기본 EndPoint

- auditevents: 호출된 Audit 이벤트 정보를 표시하는데 AuditEventRepository 빈이 필요
- beans: 애플리케이션에 있는 모든 스프링 빈 리스트를 표시
- caches: 사용 가능한 캐시를 표시
- conditions: 자동 구성 조건 내역을 생성
- configprops: @Configurationproperties의 속성 리스트를 표시
- env: 애플리케이션에서 사용할 수 있는 환경 속성을 표시
- health: 애플리케이션의 상태 정보를 표시
- httptrace: 가장 최근에 이뤄진 100건의 요청 기록을 표시하는데 HttpTraceRepository 빈이 필요
- info: 애플리케이션의 정보를 표시
- Integrationgraph: 스프링 통합 그래프를 표시하는데 spring-integration-core 모듈에 대한 의존성을 추가해야 동작
- loggers: 애플리케이션의 로거 구성을 표시하고 수정
- metrics: 애플리케이션의 메트릭 정보를 표시
- mappings: 모든 @RequestMapping의 매핑 정보를 표시

EndPoint

❖ EndPoint

✓ 기본 EndPoint

- quartz: Quartz 스케줄러 작업에 대한 정보를 표시
- scheduledtasks: 애플리케이션에서 예약된 작업을 표시
- sessions: 스프링 세션 저장소에서 사용자의 세션을 검색하고 삭제할 수 있는데 스프링 세션을 사용하는 서블릿 기반 웹 애플리케이션이 필요
- shutdown: 애플리케이션을 정상적으로 종료할 수 있는데 기본값은 비활성화 상태
- startup: 애플리케이션이 시작될 때 수집된 시작 단계 데이터를 표시하는데 BufferingApplicationStartup으로 구성된 스프링 애플리케이션이 필요
- threaddump: 스레드 덤프를 수행

EndPoint

❖ EndPoint

- ✓ Spring MVC, Spring WebFlux, Jersey를 사용하는 경우 사용할 수 있는 EndPoint
 - heapdump: 힙 덤프 파일을 리턴하는데 HotSpot VM 상에서 hprof 포맷의 파일이 반환되며 OpenJ9 JVM에서는 PHD 포맷 파일을 반환
 - jolokia: Jolokia가 클래스패스에 있을 때 HTTP를 통해 JMX 빈을 표시하는데 jolokiafore 모듈에 대한 의존성 추가가 필요하며, WebFlux에서는 사용할 수 없음
 - logfile: logging.file.name 또는 logging.file.path 속성이 설정돼 있는 경우 로그 파일의 내용을 반환
 - Prometheus: Prometheus 서버에서 스크랩할 수 있는 형식으로 메트릭을 표시하는데 micrometer-registry-prometheus 모듈의 의존성 추가가 필요

EndPoint

❖ EndPoint

- ✓ EndPoint는 활성화 여부 와 노출 여부 설정이 가능
 - 활성화는 기능 자체를 활성화할 것인지를 결정하는 것으로 비활성화된 EndPoint는 애플리케이션 컨텍스트에서 완전히 제거됨
 - EndPoint 를 활성화하려면 application.properties 파일에 속성을 추가
 - ## EndPoint 활성화
 - management.endpoint.shutdown.enabled=true
 - management.endpoint.caches.enabled=false
 - 액추에이터 설정을 통해 기능 활성화/비활성화가 아니라 EndPoint의 노출 여부만 설정하는 것도 가능한데 노출 여부는JMX를 통한 노출과 HTTP를 통한 노출이 있음
 - ## EndPoint 노출 설정
 - ## HTTP 설정
 - management.endpoints.web.exposure.include=*
 - management.endpoints.web.exposure.exclude=threaddump,heapdump
 - ## JMX 설정
 - management.endpoints.jmx.exposure.include=*
 - management.endpoints.jmx.exposure.exclude=threaddump, heapdump

EndPoint

❖ EndPoint

✓ EndPoint 노출 설정 기본 값

ID	JMX	WEB
auditevents	O	X
beans	O	X
caches	O	X
conditions	O	X
configprops	O	X
env	O	X
flyway	O	X
health	O	O
heapdump	해당 없음	X
httptrace	O	X
info	O	X
integrationgraph	O	X
jolokia	해당 없음	X
logfile	해당 없음	X
loggers	O	X
liquibase	O	X
metrics	O	X
mappings	O	X

EndPoint

❖ EndPoint

- ✓ EndPoint 노출 설정 기본 값

ID	JMX	WEB
Prometheus	해당 없음	
quartz	O	X
scheduledtasks	O	X
sessions	O	X
shutdown	O	X
startup	O	X
threaddump	O	X

Actuator 기능 확인

❖ 기본 정보

- ✓ application.properties 파일을 삭제하고 application.yml 파일을 만들고 설정 추가

server:

port: 80

management:

엔드포인트 노출 설정

endpoints:

web:

exposure:

include: "*"

Actuator 기능 확인

❖ 기본 정보

- ✓ path 확인: localhost/actuator

← → ↻ ⓘ localhost/actuator



```
{
  "_links": {
    "self": {
      "href": "http://localhost/actuator",
      "templated": false
    },
    "beans": {
      "href": "http://localhost/actuator/beans",
      "templated": false
    },
    "caches-cache": {
      "href": "http://localhost/actuator/caches/{cache}",
      "templated": true
    },
    "caches": {
      "href": "http://localhost/actuator/caches",
      "templated": false
    },
    "health": {
      "href": "http://localhost/actuator/health",
      "templated": false
    },
    "health-path": {
      "href": "http://localhost/actuator/health/{*path}",
      "templated": true
    },
    "info": {
      "href": "http://localhost/actuator/info",
      "templated": false
    },
    "conditions": {
      "href": "http://localhost/actuator/conditions",
      "templated": false
    },
    "configprops-prefix": {
      "href": "http://localhost/actuator/configprops/{prefix}",
      "templated": true
    },
    "configprops": {
      "href": "http://localhost/actuator/configprops",
      "templated": false
    },
    "env": {
      "href": "http://localhost/actuator/env",
      "templated": false
    },
    "env-toMatch": {
      "href": "http://localhost/actuator/env/{toMatch}",
      "templated": true
    },
    "loggers": {
      "href": "http://localhost/actuator/loggers",
      "templated": false
    },
    "loggers-name": {
      "href": "http://localhost/actuator/loggers/{name}",
      "templated": true
    },
    "heapdump": {
      "href": "http://localhost/actuator/heapdump",
      "templated": false
    },
    "threaddump": {
      "href": "http://localhost/actuator/threaddump",
      "templated": false
    },
    "metrics": {
      "href": "http://localhost/actuator/metrics",
      "templated": false
    },
    "metrics-requiredMetricName": {
      "href": "http://localhost/actuator/metrics/{requiredMetricName}",
      "templated": true
    },
    "scheduledtasks": {
      "href": "http://localhost/actuator/scheduledtasks",
      "templated": false
    },
    "mappings": {
      "href": "http://localhost/actuator/mappings",
      "templated": false
    }
  }
}
```

Actuator 기능 확인

❖ 기본 정보

✓ health 상태 확인: 상세 내역 확인 – 인프라 설정까지 출력

- application.yml 파일에 설정 추가

management:

endpoint:

health:

show-details: always

← → ↻ ⓘ localhost/actuator/health



```
{"status": "UP", "components": {"diskSpace": {"status": "UP", "details": {"total": 1000240963584, "free": 803881410560, "threshold": 10485760, "path": "/Users/adam/Documents/lecture/source/spring/boot/actuator/.", "exists": true}}, "ping": {"status": "UP"}}
```


Actuator 기능 확인

❖ 기본 정보

✓ bean 정보 확인 - beans

```
localhost/actuator/beans

{"contexts":{"application":{"beans":{"endpointCachingOperationInvokerAdvisor":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.actuate.endpoint.invoker.cache.CachingOperationInvokerAdvisor","resource":"class path resource
[org.springframework.boot.actuate.autoconfigure.endpoint.EndpointAutoConfiguration.class]","dependencies":
["org.springframework.boot.actuate.autoconfigure.endpoint.EndpointAutoConfiguration","environment"]},"defaultServletHandlerMapping":{"aliases":
[],"scope":"singleton","type":"org.springframework.web.servlet.HandlerMapping","resource":"class path resource
[org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfiguration$EnableWebMvcConfiguration.class]","dependencies":
["org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfiguration$EnableWebMvcConfiguration"]},"applicationTaskExecutor":{"aliases":
["taskExecutor"],"scope":"singleton","type":"org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor","resource":"class path resource
[org.springframework.boot.autoconfigure.task.TaskExecutionAutoConfiguration.class]","dependencies":
["org.springframework.boot.autoconfigure.task.TaskExecutionAutoConfiguration","taskExecutorBuilder"]},"observationRegistryPostProcessor":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.actuate.autoconfigure.observation.ObservationRegistryPostProcessor","resource":"class path resource
[org.springframework.boot.actuate.autoconfigure.observation.ObservationAutoConfiguration.class]","dependencies":{}},"characterEncodingFilter":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.web.servlet.filter.OrderedCharacterEncodingFilter","resource":"class path resource
[org.springframework.boot.autoconfigure.web.servlet.HttpEncodingAutoConfiguration.class]","dependencies":
["org.springframework.boot.autoconfigure.web.servlet.HttpEncodingAutoConfiguration"]},"forceAutoProxyCreatorToUseClassProxying":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.autoconfigure.aop.AopAutoConfiguration$ClassProxyingConfiguration$Lambda$465/Ox0000000800f19468","resource":"class path
resource [org.springframework.boot.autoconfigure.aop.AopAutoConfiguration$ClassProxyingConfiguration.class]","dependencies":{}},"healthEndpointGroups":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.actuate.autoconfigure.health.AutoConfiguredHealthEndpointGroups","resource":"class path resource
[org.springframework.boot.actuate.autoconfigure.health.HealthEndpointConfiguration.class]","dependencies":
["org.springframework.boot.actuate.autoconfigure.health.HealthEndpointConfiguration","org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@
5a059268","management.endpoint.health-org.springframework.boot.actuate.autoconfigure.health.HealthEndpointProperties"]},"management.endpoint.health-
org.springframework.boot.actuate.autoconfigure.health.HealthEndpointProperties":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.actuate.autoconfigure.health.HealthEndpointProperties","dependencies":{}},"webEndpointDiscoverer":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.actuate.autoconfigure.endpoint.web.annotation.WebEndpointDiscoverer","resource":"class path resource
[org.springframework.boot.actuate.autoconfigure.endpoint.web.WebEndpointAutoConfiguration.class]","dependencies":
["org.springframework.boot.actuate.autoconfigure.endpoint.web.WebEndpointAutoConfiguration","endpointOperationParameterMapper","endpointMediaTypes"],"org.springframework.boot.autoco
nfigure.web.servlet.MultipartAutoConfiguration":{"aliases":
[],"scope":"singleton","type":"org.springframework.boot.autoconfigure.web.servlet.MultipartAutoConfiguration","dependencies":["spring.servlet.multipart-
org.springframework.boot.autoconfigure.web.servlet.MultipartProperties"]},"org.springframework.boot.autoconfigure.web.servlet.DispatcherServletAutoConfiguration$DispatcherServletRegi
strationConfiguration":{"aliases":
```

Actuator 기능 확인

❖ 기본 정보

✓ 자동 설정 내역 - conditions

```
localhost/actuator/conditions

{"contexts":{"application":{"positiveMatches":{"AuditEventsEndpointAutoConfiguration":{"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.jmx.exposure' property"}}, "BeansEndpointAutoConfiguration":{"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.jmx.exposure' property"}}, "BeansEndpointAutoConfiguration#beansEndpoint":{"condition":"OnBeanCondition","message":"@ConditionalOnMissingBean (types: org.springframework.boot.actuate.beans.BeansEndpoint; SearchStrategy: all) did not find any beans"}}, "CachesEndpointAutoConfiguration":{"condition":"OnClassCondition","message":"@ConditionalOnClass found required class 'org.springframework.cache.CacheManager'"}, {"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.jmx.exposure' property"}}, "CachesEndpointAutoConfiguration#cachesEndpoint":{"condition":"OnBeanCondition","message":"@ConditionalOnMissingBean (types: org.springframework.boot.actuate.cache.CachesEndpoint; SearchStrategy: all) did not find any beans"}}, "CachesEndpointAutoConfiguration#cachesEndpointWebExtension":{"condition":"OnBeanCondition","message":"@ConditionalOnBean (types: org.springframework.boot.actuate.cache.CachesEndpoint; SearchStrategy: all) found bean 'cachesEndpoint'; @ConditionalOnMissingBean (types: org.springframework.boot.actuate.cache.CachesEndpointWebExtension; SearchStrategy: all) did not find any beans"}, {"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.web.exposure' property"}}, "ConditionsReportEndpointAutoConfiguration":{"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.jmx.exposure' property"}}, "ConditionsReportEndpointAutoConfiguration#conditionsReportEndpoint":{"condition":"OnBeanCondition","message":"@ConditionalOnMissingBean (types: org.springframework.boot.actuate.autoconfigure.condition.ConditionsReportEndpoint; SearchStrategy: current) did not find any beans"}}, "ConfigurationPropertiesReportEndpointAutoConfiguration":{"condition":"OnAvailableEndpointCondition","message":"@ConditionalOnAvailableEndpoint marked as exposed by a 'management.endpoints.jmx.exposure' property"}}, "ConfigurationPropertiesReportEndpointAutoConfiguration#configurationPropertiesReportEndpoint":{"condition":"OnBeanCondition","message":"@ConditionalOnMissingBean (types: org.springframework.boot.actuate.context.properties.ConfigurationPropertiesReportEndpoint; SearchStrategy: all) did not find any beans"}}, "ConfigurationPropertiesReportEndpointAutoConfiguration#configurationPropertiesReportEndpointWebExtension":{"condition":"OnBeanCondition","message":"@ConditionalOnBean (types: org.springframework.boot.actuate.context.properties.ConfigurationPropertiesReportEndpoint; SearchStrategy: all) found bean 'configurationPropertiesReportEndpoint'; @ConditionalOnMissingBean (types: org.springframework.boot.actuate.context.properties.ConfigurationPropertiesReportEndpointWebExtension; SearchStrategy: all) did not find any beans"}},
```

Actuator 기능 확인

❖ 기본 정보

✓ 환경변수 정보 - env

```
{
  "activeProfiles": [],
  "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": "*****"
        },
        "java.vendor.url": {
          "value": "*****"
        },
        "catalina.useNaming": {
          "value": "*****"
        },
        "user.timezone": {
          "value": "*****"
        },
        "java.vm.specification.version": {
          "value": "*****"
        },
        "os.name": {
          "value": "*****"
        },
        "sun.java.launcher": {
          "value": "*****"
        },
        "user.country": {
          "value": "*****"
        },
        "java.boot.library.path": {
          "value": "*****"
        },
        "spring.application.admin.enabled": {
          "value": "*****"
        },
        "sun.java.command": {
          "value": "*****"
        },
        "com.sun.management.jmxremote": {
          "value": "*****"
        },
        "http.nonProxyHosts": {
          "value": "*****"
        },
        "jdk.debug": {
          "value": "*****"
        },
        "spring.liveBeansView.mbeanDomain": {
          "value": "*****"
        },
        "sun.cpu.endian": {
          "value": "*****"
        },
        "user.home": {
          "value": "*****"
        },
        "user.language": {
          "value": "*****"
        },
        "java.specification.vendor": {
          "value": "*****"
        },
        "java.version.date": {
          "value": "*****"
        },
        "java.home": {
          "value": "*****"
        },
        "spring.output.ansi.enabled": {
          "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": "*****"
        },
        "sun.management.compiler": {
          "value": "*****"
        },
        "ftp.nonProxyHosts": {
          "value": "*****"
        },
        "java.runtime.version": {
          "value": "*****"
        },
        "user.name": {
          "value": "*****"
        },
        "spring.jmx.enabled": {
          "value": "*****"
        },
        "path.separator": {
          "value": "*****"
        },
        "management.endpoints.jmx.exposure.include": {
          "value": "*****"
        },
        "os.version": {
          "value": "*****"
        },
        "java.runtime.name": {
          "value": "*****"
        },
        "file.encoding": {
          "value": "*****"
        },
        "java.vm.name": {
          "value": "*****"
        },
        "java.vendor.url.bug": {
          "value": "*****"
        },
        "java.io.tmpdir": {
          "value": "*****"
        },
        "catalina.home": {
          "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.version": {
          "value": "*****"
        },
        "java.rmi.server.randomIDs": {
          "value": "*****"
        },
        "sun.io.unicode.encoding": {
          "value": "*****"
        },
        "socksNonProxyHosts": {
          "value": "*****"
        },
        "java.class.version": {
          "value": "*****"
        }
      }
    },
    {
      "name": "systemEnvironment",
      "properties": {
        "_CFBundleIdentifier": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "PATH": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "SHELL": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_DEFAULT_ENV": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_EXE": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_PYTHON_EXE": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "OLDPWD": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "USER": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "COMMAND_MODE": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "TMPDIR": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_PREFIX": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "SSH_AUTH_SOCK": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CE_CONDA": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "XPC_FLAGS": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CE_M": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "JAVA_MAIN_CLASS_32967": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "_CF_USER_TEXT_ENCODING": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_PROMPT_MODIFIER": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "LOGNAME": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "LC_CTYPE": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "PWD": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "XPC_SERVICE_NAME": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "CONDA_SHLVL": {
          "value": "*****",
          "origin": "System Environment Property"
        },
        "HOME": {
          "value": "*****",
          "origin": "System Environment Property"
        }
      }
    },
    {
      "name": "Config resource 'class path resource [application.yml]' via location 'optional:classpath:/'",
      "properties": {
        "server.port": {
          "value": "*****",
          "origin": "class path resource [application.yml] - 2:9"
        },
        "management.endpoints.web.exposure.include": {
          "value": "*****",
          "origin": "class path resource [application.yml] - 9:18"
        },
        "management.endpoint.health.show-details": {
          "value": "*****",
          "origin": "class path resource [application.yml] - 13:21"
        },
        "devtools": {
          "value": "*****",
          "origin": "spring.groovy.template.cache"
        },
        "server.servlet.jsp.init-parameters.development": {
          "value": "*****",
          "origin": "spring.mvc.log-resolved-exception"
        },
        "spring.web.resources.chain.cache": {
          "value": "*****",
          "origin": "spring.thymeleaf.cache"
        },
        "spring.freemarker.cache": {
          "value": "*****",
          "origin": "spring.error.include-stacktrace"
        },
        "spring.reactor.netty.shutdown-quiet-period": {
          "value": "*****",
          "origin": "spring.graphql.graphql.enabled"
        },
        "spring.h2.console.enabled": {
          "value": "*****",
          "origin": "spring.servlet.session.persistent"
        },
        "server.error.include-binding-errors": {
          "value": "*****",
          "origin": "spring.template.provider.cache"
        },
        "server.error.include-message": {
          "value": "*****",
          "origin": "spring.mustache.servlet.cache"
        },
        "spring.web.resources.cache.period": {
          "value": "*****",
          "origin": "*****"
        }
      }
    }
  ]
}
```


❖ 기본 정보

- ✓ 로거 정보 - loggers

← → ↻ ⓘ localhost/actuator/loggers

```
{ "levels": { "OFF", "ERROR", "WARN", "INFO", "DEBUG", "TRACE" }, "loggers": { "ROOT": { "configuredLevel": "INFO", "effectiveLevel": "INFO", "org": { "effectiveLevel": "INFO", "org.springframework": { "effectiveLevel": "INFO", "org.springframework.web": { "effectiveLevel": "INFO", "org.springframework.web.servlet": { "effectiveLevel": "INFO", "org.springframework.web.servlet.HandlerMapping": { "effectiveLevel": "INFO", "org.springframework.web.servlet.HandlerMapping.Mappings": { "effectiveLevel": "INFO", "com": { "effectiveLevel": "INFO", "com.adamssoft": { "effectiveLevel": "INFO", "com.adamssoft.actuator": { "effectiveLevel": "INFO", "com.adamssoft.actuator.ActuatorApplication": { "effectiveLevel": "INFO", "io": { "effectiveLevel": "INFO", "io.micrometer": { "effectiveLevel": "INFO", "io.micrometer.common": { "effectiveLevel": "INFO", "io.micrometer.common.util": { "effectiveLevel": "INFO", "io.micrometer.common.util.internal": { "effectiveLevel": "INFO", "io.micrometer.common.util.internal.logging": { "effectiveLevel": "INFO", "io.micrometer.common.util.internal.logging.InternalLoggerFactory": { "effectiveLevel": "INFO", "io.micrometer.core": { "effectiveLevel": "INFO", "io.micrometer.core.instrument": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.binder": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.binder.jvm": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.binder.jvm.ExecutorServiceMetrics": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.binder.jvm.JvmGcMetrics": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.internal": { "effectiveLevel": "INFO", "io.micrometer.core.instrument.internal.DefaultGauge": { "effectiveLevel": "INFO", "org": { "effectiveLevel": "INFO", "org.apache": { "effectiveLevel": "INFO", "org.apache.catalina": { "effectiveLevel": "INFO", "org.apache.catalina.core": { "effectiveLevel": "INFO", "org.apache.catalina.core.ContainerBase": { "effectiveLevel": "INFO", "org.apache.catalina.core.ContainerBase.[Tomcat]": { "effectiveLevel": "INFO", "org.apache.catalina.core.ContainerBase.[Tomcat].[localhost]": { "effectiveLevel": "INFO", "org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/]": { "effectiveLevel": "INFO", "org.apache.catalina.core.StandardEngine": { "effectiveLevel": "INFO", "org.apache.catalina.core.StandardService": { "effectiveLevel": "INFO", "org.apache.catalina.startup": { "effectiveLevel": "INFO", "org.apache.catalina.startup.DigesterFactory": { "configuredLevel": "ERROR", "effectiveLevel": "ERROR", "org.apache.catalina.util": { "effectiveLevel": "INFO", "org.apache.catalina.util.LifecycleBase": { "configuredLevel": "ERROR", "effectiveLevel": "ERROR", "org.apache.coyote": { "effectiveLevel": "INFO", "org.apache.coyote.http11": { "effectiveLevel": "INFO", "org.apache.coyote.http11.Http11NioProtocol": { "configuredLevel": "WARN", "effectiveLevel": "WARN", "org.apache.sshd": { "effectiveLevel": "INFO", "org.apache.sshd.common": { "effectiveLevel": "INFO", "org.apache.sshd.common.util": { "effectiveLevel": "INFO", "org.apache.sshd.common.util.SecurityUtils": { "configuredLevel": "WARN", "effectiveLevel": "WARN", "org.apache.tomcat": { "effectiveLevel": "INFO", "org.apache.tomcat.util": { "effectiveLevel": "INFO", "org.apache.tomcat.util.net": { "effectiveLevel": "INFO", "org.apache.tomcat.util.net.NioSelectorPool": { "configuredLevel": "WARN", "effectiveLevel": "WARN", "org.eclipse": { "effectiveLevel": "INFO", "org.eclipse.jetty": { "effectiveLevel": "INFO", "org.eclipse.jetty.util":
```

Custom 기능 확인

❖ 정보 제공 인터페이스의 구현체

- ✓ 커스텀 기능을 설정할 때는 별도의 구현체 클래스를 작성해서 내용을 추가하는 방법을 활용
- ✓ InfoContributor 인터페이스를 제공하고 있는데 이 인터페이스를 구현하는 클래스를 생성



Custom 기능 확인

❖ 정보 제공 인터페이스의 구현체

- ✓ 구현체 클래스 작성 – config.CustomInfoContributor

@Component

public class CustomInfoContributor implements InfoContributor {

@Override

public void contribute(Builder builder) {

Map<String, Object> content = new HashMap<>();

content.put("code-info", "InfoContributor 구현체에서 정의한 정보입니다.");

builder.withDetail("custom-info-contributor", content);

}

}

Custom 기능 확인

- ❖ 정보 제공 인터페이스의 구현체
 - ✓ 확인

```
{"custom-info-contributor":{"code-info":"InfoContributor 구현체에서 정의한 정보입니다."}}
```

Custom 기능 확인

❖ Custom EndPoint

✓ EndPoint 클래스 생성

```
@Component
@endpoint(id = "note", enableByDefault = true)
public class NoteEndpoint {
    private Map<String, Object> noteContent = new HashMap<>();

    @ReadOperation
    public Map<String, Object> getNote(){
        return noteContent;
    }
    @WriteOperation
    public Map<String, Object> writeNote(String key, Object value){
        noteContent.put(key,value);
        return noteContent;
    }
    @DeleteOperation
    public Map<String, Object> deleteNote(String key){
        noteContent.remove(key);
        return noteContent;
    }
}
```


Custom 기능 확인

❖ Custom EndPoint

✓ note 확인

POST http://localhost/actuator/note

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "key": "description",
3   "value": "설명 부분을 기입합니다."
4 }
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 20 ms Size: 243 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "description": "설명 부분을 기입합니다."
3 }
```

Custom 기능 확인

❖ Custom EndPoint

✓ note 확인

← → ↻ ⓘ localhost/actuator/note

```
{"description": "설명 부분을 기입합니다."}
```

Custom 기능 확인

❖ Custom EndPoint

✓ note 확인

The screenshot displays a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost/actuator/note?key=description
- Body:** A JSON object with the following structure:

```
{
  "key": "description",
  "value": "설명 부분을 기입합니다."
}
```
- Response:** Status: 200 OK, Time: 8 ms, Size: 194 B. The response body is shown in a 'Pretty' JSON format.