

Mastering Spring Boot's Actuator

Madhura Bhave

@madhurabhave23

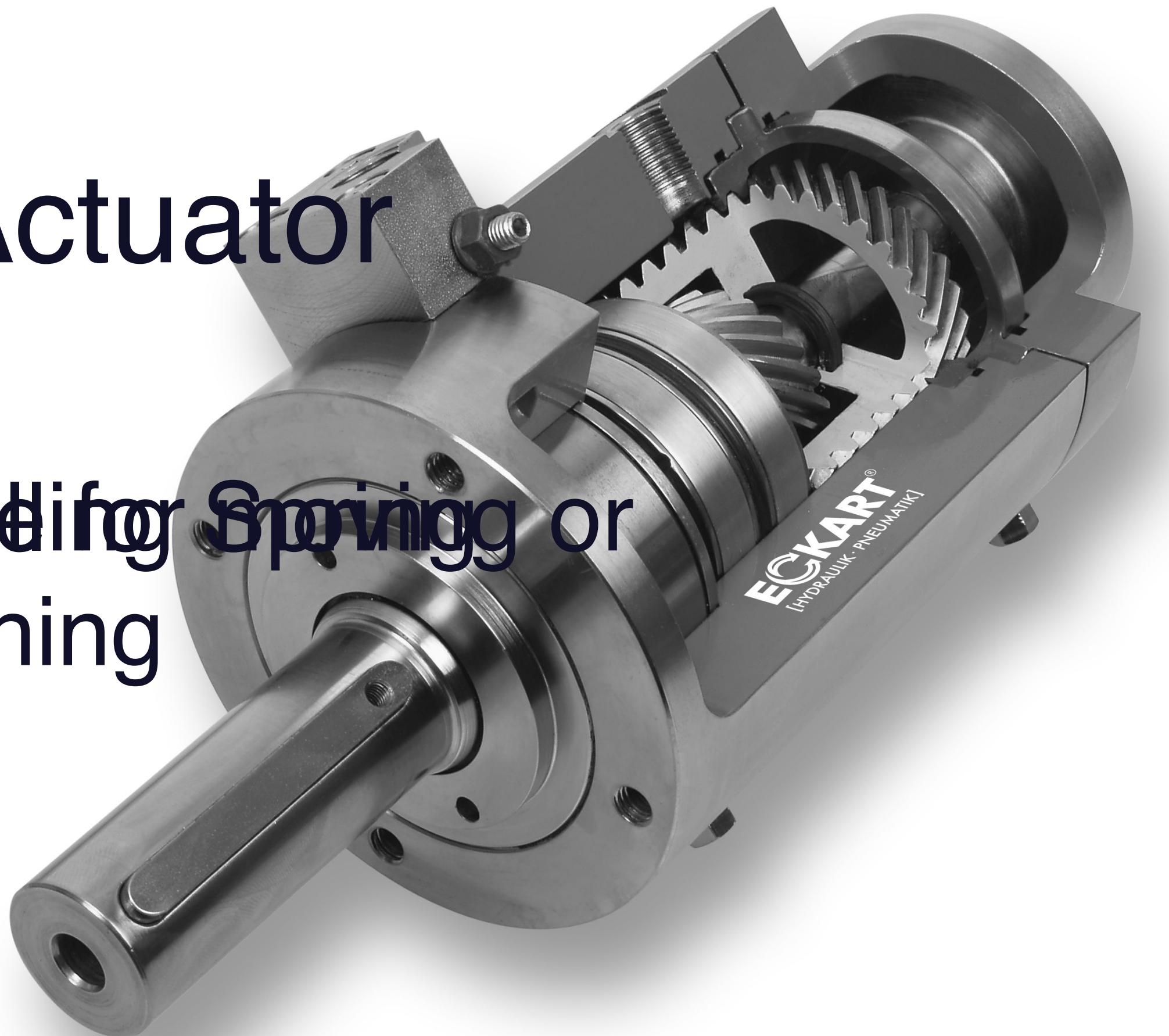
Actuator?

Actuator?

~~S~~trutagoBoot Actuator

noun

~~mədbrænfələdəntrɪdɪfɔr~~ Spring or
~~Bəotrælfibjɛsdiɒrɛshɪŋ~~



What's new?

What's new?

- Endpoints under /actuator
- management.endpoints.web.base-path
- management.endpoints.web.exposure.include
management.endpoints.web.exposure.exclude
- management.endpoint.shutdown.enabled
- management.endpoint.health.show-details
- EndpointRequest
- Jersey, WebFlux, or Spring MVC

What's new?

- New endpoints
 - ScheduledTasksEndpoint
 - SessionsEndpoint
 - CachesEndpoint
 - IntegrationGraphEndpoint

/actuator/health/{component}/{instance}

```
{  
  "status" : "UP",  
  "details" : {  
    "db" : {  
      "status" : "UP",  
      "details" : {  
        "database" : "HSQL Database Engine",  
        "hello" : 1  
      }  
    },  
    "broker" : {  
      "status" : "UP",  
      "details" : {  
        "us1" : {  
          "status" : "UP",  
          "details" : {  
            "version" : "1.0.2"  
          }  
        },  
        "us2" : {  
          "status" : "UP",  
          "details" : {  
            "version" : "1.0.4"  
          }  
        }  
      }  
    }  
  }  
}
```

CloudFoundry Integration

CloudFoundry Integration

- Endpoints available under /cloudfoundryapplication
- Requires a valid UAA token
- management.cloudfoundry.enabled
- Jersey, WebFlux, or Spring MVC

Web API documentation

Metrics



Writing your own endpoints

```
/**  
 * {@link Endpoint} to expose a user's {@link Session}s.  
 *  
 * @author Vedran Pavic  
 * @since 2.0.0  
 */  
@Endpoint(id = "sessions")  
public class SessionsEndpoint {  
  
    // ...  
  
}
```

```
@ReadOperation
public SessionsReport sessionsForUsername(String username) {
    Map<String, ? extends Session> sessions =
        this.sessionRepository.findByIndexNameAndIndexValue(
            FindByIndexNameSessionRepository.PRINCIPAL_NAME_INDEX_NAME,
            username);
    return new SessionsReport(sessions);
}
```

```
@ReadOperation
public SessionDescriptor getSession(@Selector String sessionId) {
    Session session = this.sessionRepository.findById(sessionId);
    if (session == null) {
        return null;
    }
    return new SessionDescriptor(session);
}
```

```
@DeleteOperation  
public void deleteSession(@Selector String sessionId) {  
    this.sessionRepository.deleteById(sessionId);  
}
```

```
@Endpoint(id = "loggers")
public class LoggersEndpoint {

    @WriteOperation
    public void configureLogLevel(@Selector String name,
        @Nullable LogLevel configuredLevel) {
        Assert.notNull(name, "Name must not be empty");
        this.loggingSystem.setLogLevel(name, configuredLevel);
    }

}
```

Writing your own endpoints

Beyond the basics

```
@EndpointWebExtension(endpoint = HealthEndpoint.class)
public class HealthEndpointWebExtension {

    @ReadOperation
    public WebEndpointResponse<Health> getHealth(
        SecurityContext securityContext) {
        return this.responseMapper.map(
            this.delegate.health(), securityContext);
    }

}
```

SpringOne Platform

by Pivotal.

Thanks!

Madhura Bhave

@madhurabhave23