Method-Level Security versus Filter Chains

Combine OAuth2 (OpenID Connect)

with Spring @PreAuthorize

Abstract

Servlet filters are a Java standard dating back to the beginning of Java. They, along with Servlets and the Catalina Servlet Container are an integral part of Tomcat.

This document covers an application that turns-off any

ServletFilter interposition of HTTP Security policies and allows

method-level, role-based security policies to be applied

Tomcat is the reference-standard implementation of the Servlet 3.0 Standards/Specs. Any Web Server purporting to be compliant with the Servlet 3.0 Standards/Specs must exhibit the same capabilities as does Tomcat.

Additionally, Tomcat is the default Web Server engine powering Spring boot Applications, with ServletFilters acting as container-wide interceptors for Servlets – ServletFilters fire both before *and after* Servlets.

Spring’s DispatcherServlet is not new, nor is it exempt from the architecture described by the Servlet 3.0 Specification.

And, since “Controllers” are just adjuncts to Servlets/Filters, affording a specificity to Servlets, they to, comply with the HTTP Security policies which are inherent in all Servlet-based applications.

This means that any security policies inforce at the (typically) Filter level supersede (override) security policies applied at the Controller level. By extension, member methods of any Controller are subject to the same ServletFilter security policies.

This document covers an application that turns-off any

ServletFilter interposition of HTTP Security policies and allows

method-level, role-based security policies to be applied

****