

Spring Security Filter Zinciri ve Filter Mimarisi (İleri Düzey)



Spring Security Filter

JAVA Eğitimleri

Zinciri

Authentication token'ın request'ler arasında HttpSession'da saklanmasını şağlar

HTTP Request

ChannelProcessing Filter



SecurityContext PersistenceFilter



ConcurrentSession Filter



CSRF token yönetimini gerçekleştirir HeaderWriter Filter



CsrfFilter

www.java-egitimleri.com

Kullanıcının eş zamanlı ve süresi dolmuş oturumlarını

denetler

Web isteğinin HTTPS'den

gelmesini garanti eder

X-Frame-Options, X-XSS-Protection, X-Content-Type-Options gibi headerları set ederek tarayıcı korumasının devreye alınmasını sağlar

Spring Security Filter



Zinciri

Logout işlemini gerçekleştirir

Web isteğinin getRemoteUser(), isUserInRole() metodlarında SecurityContext'i kullanmasını sağlar LogoutFilter

UsernamePassword Authentication Filter

BasicAuthentication Filter

SecurityContext HolderAware RequestFilter

RememberMe Authentication Filter

www.java-egitimleri.com

Form tabanlı authentication yapar

HTTP Basic authentication yapar

Web isteğinde remember-me cookie mevcutsa, değerini SecurityContextH older'a set eder

Spring Security Filter Zinciri

Authentication Exeption ve AccessDeniedE xception'ların uygun sayfalara yönlendirilmesi ni sağlar

> Hedef URL'e yetkiye göre erişilmesini sağlar. **AuthenticationExc** eption veya **AccessDeniedExce** ption fırlatabilir



Anonymous AuthenticationFilter



Session ManagementFilter



ExceptionTranslation Filter



FilterSecurity Interceptor



secureWebPage.jsp

www.java-egitimleri.com

SecurityContext Holder'a anonymous **Authentication** token'ı koyar

JAVA Eğitimleri

Session fixation saldırısına karşı önlem alır, eş zamanlı oturumları denetler

Container'da Birden Fazla <security:http> Elemanı



- Spring konfigürasyon dosyasında aynı anda birden fazla <security:http> elemanı tanımlanabilir
- Bunun amacı uygulama içerisinde farklı
 URI pattern'larını farklı kimliklendirme metotları ile ele almaktır
- Ya da her bir URI pattern'ında farklı güvenlik kontrollerini devreye sokmaktır

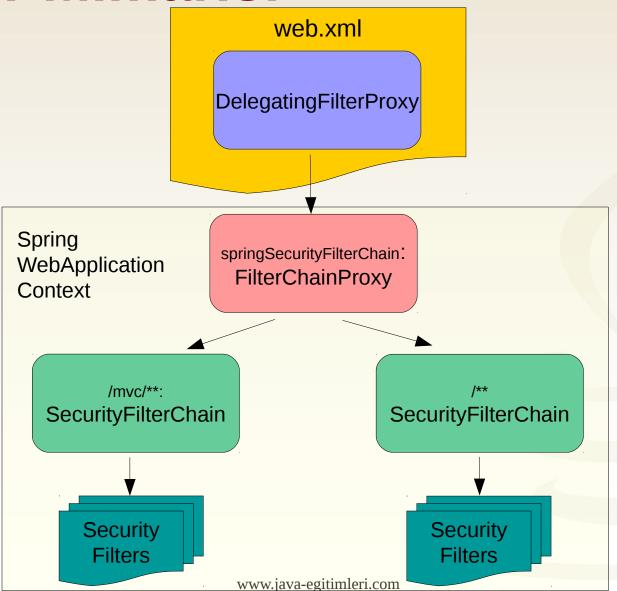
Container'da Birden Fazla <security:http> Elemanı



```
<beans...>
    <security:http pattern="/mvc/**" create-session="stateless">
        <security:http-basic />
        <security:intercept-url pattern="/**" access="hasRole('ROLE_USER')"/>
   </security:http>
    <security:http pattern="/**" auto-config="true">
        <security:access-denied-handler</pre>
                error-page="/accessDenied.jsp"/>
        <security:form-login login-page="/login.jsp"</pre>
            authentication-failure-url="/login.jsp?error=true" />
        <security:intercept-url pattern="/login.jsp"</pre>
            access="isAnonymous()" />
        <security:intercept-url pattern="/editor.jsp"</pre>
            access="hasRole('ROLE EDITOR')" />
        <security:intercept-url pattern="/**"</pre>
            access="hasRole('ROLE_USER')" />
    </security:http>
</beans>
```

Spring Security Filter Mimarisi





Spring Security Filter Mimarisi



- Her bir http elemanı ayrı bir
 SecurityFilterChain tipinde bean'e karşılık gelir
- Ayrıca herhangi bir http elemanının tanımlanması ApplicationContext'de
 FilterChainProxy tipinde tek bir bean tanımlanmasını da tetikler
- SecurityFilterChain bean'leri bu
 FilterChainProxy bean'i içerisinde yer alırlar

Spring Security Filter Mimarisi



- FilterChainProxy tipindeki bean'in ismi "springSecurityFilterChain" dir ve değiştirilemez
- web.xml'de tanımlı DelegatingFilterProxy tipindeki filter ApplicationContext'de bu isimde bir bean'i arar
- Fakat her bir SecurityFilterChain bean'ine http elemanı üzerinden ayrı bir bean ismi atanabilir

Spring Security Debug Info



<security:debug/>

- Debug altyapısını aktive eder
- Security Filter'lara gelen request'leri okunur biçimde console'a basar
- Bu request'lerdeki header, request param, request path gibi önemli bilgilerde console'da yer alır
- Filter zincirindeki Filter'ları listeler
- Sadece development ortamı içindir

Spring Security Debug Info

AnonymousAuthenticationFilter

SessionManagementFilter ExceptionTranslationFilter FilterSecurityInterceptor



```
Request received for POST '/j spring security check':
(POST /petclinic/j spring security check)@1819769884 org.eclipse.jetty.server.Request@6c777clc
servletPath:/j spring security check
pathInfo:null
headers:
Cookie: JSESSIONID=1g3txc33ic4ny1g4ht2n29wr1v; locale=en
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86 64; rv:32.0) Gecko/20100101 Firefox/32.0
Referer: http://localhost:8080/petclinic/spring security login
Connection: keep-alive
Host: localhost:8080
Accept-Language: tr,en-us; g=0.7,en; g=0.3
Accept-Encoding: gzip, deflate
Content-Length: 44
Content-Type: application/x-www-form-urlencoded
Security filter chain: [
  SecurityContextPersistenceFilter
  WebAsyncManagerIntegrationFilter
  LogoutFilter
  UsernamePasswordAuthenticationFilter
  DefaultLoginPageGeneratingFilter
  BasicAuthenticationFilter
  RequestCacheAwareFilter
  SecurityContextHolderAwareRequestFilter
```



İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com

