Kullanıcı Oturum Yönetimi

harezmi bilişim çözümleri

Kullanıcı Oturumlarının Yönetilmesi

- Spring Security üç farklı türde kullanıcı **oturum yönetimi** sağlar
  - Session **fixation** saldırılarının önüne geçmek için login sonrasında yeni session oluşturularak işleme devam edilmesi
  - Session **timeout**'ların takip edilmesi
  - Aynı kullanıcının **farklı noktalardan oturum açma**sının yönetimi

Session Fixation Saldırılarının Önlenmesi

Alabileceği değerler:
none
newSession
migrateSession
changeSessionId

```
<security:http>

    <security:session-management
                        session-fixation-protection="changeSessionId">
        ...
    </security:session-management>

</security:http>
```

- changeSessionId **Servlet 3.1** tarafından desteklenir, Servlet 3.1 container için **default** değerdir
- İçeriği değiştirmeden **HttpServletRequest.changeSessionId()** metodu ile session id yenilenir

Session Timeout'ların Takibi

- Http oturumlarının sonlanması **web.xml**'e tanımlanacak bir **HttpSessionListener** ile takip edilebilir

```
<web-app>
    <listener>
        <listener-class>
org.springframework.security.web.session.HttpSessionEventPublisher
        </listener-class>
  </listener>
</web-app>
```

Session Timeout'ların Takibi

- Yeni bir HttpSession yaratıldığında **HttpSessionCreatedEvent** , mevcut bir HttpSession invalidate edildiğinde ise **HttpSessionDestroyedEvent** publish eder
- Spring Security altyapısı bu event'leri yakalayarak kullanıcıların **aktif oturumlarının sayısı**nı yönetir

Session Timeout'ların Takibi

- Bu listener olmadan eşzamanlı oturum kontrolü aktive edilmiş ise **kullanıcının limit aşımı durumunda** sisteme bir daha girmesi mümkün olmaz! (error-if-maximum-exceeded=true)

Eş Zamanlı Oturum Yönetimi

- Kullanıcının aynı anda birden fazla farklı yerden açabileceği **oturum sayısı** sınırlandırılabilir
- İstenirse **eski oturum sonlandırılır**, ya da **yeni oturuma izin verilmez**
- İkinci durumda, uygulama kapatılmadan tarayıcının kapandığı durumlarda kullanıcının oturumunun düşmesi için **session timeout kadar beklemek gerekebilir**

Eş Zamanlı Oturum Yönetimi

```
<security:http auto-config="true">

   <security:intercept-url pattern="/**"
access="hasRole('ROLE_USER')" />

   <security:session-management>
      <security:concurrency-control max-sessions="1"
error-if-maximum-exceeded="false" expired-url="/"/>
   </security:session-management>

</security:http>
```

error-if-maximum-exceeded:
false: eski oturumu sonlandırır, eski oturumdaki kullanıcı expired-url'e yönlendirilir
true: yeni oturum açmayı engeller, login ekranında hata mesajı gösterilir

İletişim

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