

# Spring Security ve Event Yönetimi



# Spring Security ve ApplicationEvent'ler

- Spring Security **başarılı ve başarısız her kimliklendirme ve yetkilendirme işleminin ardından** ApplicationContext'e bir event fırlatır
- **ApplicationEvent** tipindeki bu event'ler **ApplicationListener** arayüzünü implement eden bean'ler ile yakalanarak uygulamaya özel işlem yapılabilir

# Kimliklendirme Event'leri

- Hepsi **AbstractAuthenticationEvent** sınıfından türer
- Bu event üzerinden **current Authentication nesnesine** erişilebilir
- **AuthenticationSuccessEvent:**
  - Kimliklendirme işleminin başarılı olduğunu bildirir
- **AuthenticationUserSwitchEvent:**
  - User context switch olduğunu bildirir

# Kimliklendirme Event'leri

- **InteractiveAuthenticationSuccessEvent:**
  - Son kullanıcının gerçekleştirdiği başarılı bir kimliklendirme işlemi olduğunu bildirir
- **SessionFixationProtectionEvent:**
  - Session fixation koruması tarafından HTTP Session ID'de bir değişiklik gerçekleştirildiğini bildirir

# Kimliklendirme Event'leri

- Başarısız kimliklendirme işlemleri ile ilgili fırlatılan event'ler ise **AbstractAuthenticationFailureEvent** sınıfından türerler
- Bu event'ler üzerinden **current Authentication** bilgisinin yanı sıra meydana gelen **AuthenticationException**'a da erişilebilir

# Kimliklendirme Event'leri

- **AuthenticationFailureBadCredentialsEvent:**
  - Kimliklendirme işleminin geçersiz credential bilgisi nedeni ile başarısız olduğunu bildirir
- **AuthenticationFailureCredentialsExpiredEvent:**
  - Kimliklendirme işleminin kullanım süresi dolmuş (expired) credential bilgisi nedeni ile başarısız olduğunu bildirir

# Kimliklendirme Event'leri

- **AuthenticationFailureDisabledEvent:**
  - Kimliklendirme işleminin kullanıcının hesabının pasif durumda (disabled) olması nedeni ile başarısız olduğunu bildirir
- **AuthenticationFailureExpiredEvent:**
  - Kimliklendirme işleminin kullanıcının hesabının kullanım süresi dolmuş (expired) olması nedeni ile başarısız olduğunu bildirir

# Kimliklendirme Event'leri

- **AuthenticationFailureLockedEvent:**
  - Kimliklendirme işleminin kullanıcının hesabının kilitli (locked) olması nedeni ile başarısız olduğunu bildirir
- **AuthenticationFailureProviderNotFound Event:**
  - Kimliklendirme işleminin isteği ele alacak uygun bir AuthenticationProvider olmaması nedeni ile başarısız olduğunu bildirir



# Kimliklendirme Event'leri

- **AuthenticationFailureProxyUntrustedEvent:**
  - Kimliklendirme işleminin CAS user ticket bilgisinin untrusted bir proxy tarafından üretilmesi nedeni ile başarısız olduğunu bildirir
- **AuthenticationFailureServiceException Event:**
  - Kimliklendirme işleminin AuthenticationManager içerisinde meydana gelen herhangi başka bir hata nedeni ile başarısız olduğunu bildirir

# JAAS Kimliklendirme Event'leri

- **JaasAuthenticationProvider** tarafından fırlatılan kimliklendirme event'leri **JaasAuthenticationEvent** sınıfından türerler
- **JaasAuthenticationSuccessEvent:**
  - Kullanıcı başarılı biçimde LoginContext'e login olup, Callback ve AuthorityGranter nesneleri çalıştıktan sonra fırlatılır
- **JaasAuthenticationFailedEvent**
  - LoginContext.login() metodu içerisinde herhangi bir hata olduğu vakit fırlatılır

# Yetkilendirme Event'leri

- Sayfa, metot veya kayıt düzeyinde yetkilendirme işlemlerinden sonra fırlatılan event'lerdir
- **AbstractAuthorizationEvent** sınıfından türerler
- Yetkilendirmenin yapıldığı “**secure object**” **Event Source** olarak erişilebilir

# Yetkilendirme Event'leri

- **AuthenticationCredentialsNotFoundEvent:**
  - Yetkilendirme işleminin SecurityContextHolder içerisinde current Authentication bilgisine erişilemediği için başarısız olduğunu bildirir
- **AuthorizationFailureEvent:**
  - Yetkilendirme işleminin kullanıcının (principal) istek için uygun yetkiye sahip olmaması nedeni ile başarısız olduğunu bildirir

# Yetkilendirme Event'leri

- **AuthorizedEvent:**
  - Yetkilendirme işleminin başarılı biçimde gerçekleştiğini, kullanıcının secure object'e erişim yetkisi olduğunu bildirir
  - Event içinden secure object'e erişen current Authentication bilgisine ve secure object üzerinde uygulanmış ConfigAttribute bilgilerine erişilebilir
  - Secure object'e erişimden hemen önce fırlatılır

# Yetkilendirme Event'leri

- **PublicInvocationEvent:**
  - Public secure object'e erişmeden hemen önce fırlatılır
  - Public secure object, kendisi için hiç ConfigAttribute tanımlanmamış secure object'tir
  - Public secure object'e erişim herhangi bir şekilde kimliklendirme ve yetkilendirmeyi tetiklemez

## Bilgilendirme Event'leri

- Bu event'ler **HttpSessionEventPublisher** tarafından fırlatılırlar
- HttpSessionEventPublisher'ın **web.xml**'de tanımlanması gerekir
- Bu event'ler üzerinden **current HttpSession** bilgisine erişmek de mümkündür

# HTTP Oturumu

## Bilgilendirme Event'leri

- **HttpSessionCreatedEvent:**
  - HttpSession yaratıldığı vakit fırlatılır
- **HttpSessionDestroyedEvent:**
  - HttpSession sonlandırıldığı vakit fırlatılır



# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

