

# Yetkilendirme ve AOP



# Security Interceptors ile Yetkilendirme

# Security Interceptor Nedir?

- İki tür Security Interceptor mevcuttur
- AOP Alliance ile MethodInvocation'ların (spring servis bean'lerin public metotlarının) yetkilendirildiği  
**MethodSecurityInterceptor**
- AspectJ ile herhangi bir nesnenin ve metodun yetkilendirildiği  
**AspectJSecurityInterceptor**

# MethodSecurity Interceptor

- `<security:global-method-security>`'de arka tarafta MethodSecurityInterceptor'ü kullanmaktadır
- Sadece **public** metot düzeyinde yetkilendirme yapabilir
- `<security:global-method-security>` elemanı yerine **MethodSecurityInterceptor** konfigürasyonu ile sadece bir veya bir grup bean'in metotları **spesifik biçimde** yetkilendirilebilir

# MethodSecurity Interceptor

```
<bean id="petClinicServiceSecurityInterceptor"  
class="org.springframework.security.access.intercept.aopalliance.MethodSecurityInterceptor">  
    <property name="authenticationManager" ref="authenticationManager" />  
    <property name="accessDecisionManager" ref="accessDecisionManager" />  
    <property name="afterInvocationManager" ref="afterInvocationManager" />  
    <property name="securityMetadataSource">
```

```
        <security:method-security-metadata-source>  
            <security:protect  
method="com.javaegitimleri.petclinic.service.PetClinicService.delete*"  
access="ROLE_ADMIN" />  
            <security:protect  
method="com.javaegitimleri.petclinic.service.PetClinicService.find*"  
access="IS_AUTHENTICATED_FULLY" />  
        </security:method-security-metadata-source>
```

```
    </property>  
</bean>
```

```
<bean id="petClinicService"  
class="com.javaegitimleri.clinic.service.PetClinicServiceImpl">  
    <property name="petClinicDao" ref="petClinicDao"/>  
</bean>
```

# MethodSecurity Interceptor

```
<bean
class="org.springframework.aop.framework.autoproxy.Bean
NameAutoProxyCreator">
  <property name="interceptorNames">
    <list>
<value>petClinicServiceSecurityInterceptor</value>
    </list>
  </property>
  <property name="beanNames">
    <list>
      <value>petClinicService</value>
    </list>
  </property>
</bean>
```

# AspectJSecurity

## Interceptor

- **<security:global-method-security>**'de yetkilendirilecek metotların bulunduğu nesneler **Spring managed bean** olmalıdır
- **AspectJSecurityInterceptor**'de ise metotları intercept edilecek nesnelerin **Spring managed bean** olmasına **gerek yoktur**
- **Domain instance düzeyinde** yetkilendirme için de kullanılabilir
- Aspect dokuma işleminde **AspectJ compiler** kullanılır

# AspectJSecurity Interceptor

```
<bean id="petClinicServiceSecurityInterceptor"  
class="org.springframework.security.access.intercept.aspectj.AspectJMethodSecurityInterceptor">  
    <property name="authenticationManager" ref="authenticationManager" />  
    <property name="accessDecisionManager" ref="accessDecisionManager" />  
    <property name="afterInvocationManager"  
ref="afterInvocationManager" />  
    <property name="securityMetadataSource">
```

```
        <security:method-security-metadata-source>  
            <security:protect  
method="com.javaegitimleri.petclinic.service.PetClinicService.delete*"  
access="ROLE_ADMIN" />  
            <secsecurityprotect  
method="com.javaegitimleri.petclinic.service.PetClinicService.find*"  
access="IS_AUTHENTICATED_FULLY" />  
        </security:method-security-metadata-source>
```

```
    </property>  
</bean>
```



# AspectJSecurity Interceptor

```
public aspect PetClinicSecurityAspect {
    private AspectJMethodSecurityInterceptor securityInterceptor;

    pointcut petClinicServiceExecution():
        execution(public * *(..)) &&
        within(com.javaegitimleri.petclinic.service.PetClinicService);

    Object around(): petClinicServiceExecution() {
        if (this.securityInterceptor == null) {
            return proceed();
        }

        AspectJCallback callback = new AspectJCallback() {
            public Object proceedWithObject() {
                return proceed();
            }
        };

        return this.securityInterceptor.invoke(thisJoinPoint, callback);
    }

    public void setSecurityInterceptor(
        AspectJMethodSecurityInterceptor securityInterceptor) {
        this.securityInterceptor = securityInterceptor;
    }
}
```

# AspectJSecurity Interceptor

```
<bean id="petClinicService"  
class="com.javaegitimleri.clinic.service.PetClinicServiceImpl">  
<property name="petClinicDao" ref="petClinicDao"/>  
</bean>
```

```
<bean  
class="com.javaegitimleri.clinic.controller.PetClinicSecurityAspect"  
factory-method="aspectOf">  
<property name="securityInterceptor"  
ref="petClinicServiceSecurityInterceptor"/>  
</bean>
```

```
<context:load-time-weaver/>
```

→ /META-INF/aop.xml içerisinde  
PetClinicServiceAspect enable edilmelidir

JVM -javaagent:/path/to/spring-instrument.jar  
argümanı ile çalıştırılmalıdır

# Protect-pointcut İfadesi ile Yetkilendirme

- Security annotasyonlarına **alternatiftir**
- **XML** konfigürasyonu içerisinde pointcut ifadesi ile **hangi metotlara ne yetki** ile erişilebileceğini tanımlar

```
<security:global-method-security secured-  
annotations="enabled">  
    <security:protect-pointcut expression="execution(*  
com.javaegitimleri.petclinic.service.*Service.*(..))"  
access="ROLE_USER" />  
</security:global-method-security>
```

# İletişim

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

