

Metot Düzeyinde Yetkilendirme



Metot Erişiminin Yetkilendirilmesi

- Spring tarafından yönetilen bean'lerin metotları **invoke edilmeden önce veya sonra yetki kontrolü** yapılabilir
- Metot argümanlarının ve return değerlerinin **filtrelenmesi** de mümkündür (ACL)
- Devreye girmesi için aşağıdaki elemanın tanımlanması yeterlidir

```
<security:global-method-security  
    pre-post-annotations="enabled"/>
```

Pre-Post Annotasyonlar ile Metot Erişiminin Yetkilendirilmesi

- PreAuthorize, PostAuthorize, PreFilter ve PostFilter annotasyonları ile yetkilendirme yapılır
- Metoda erişimden önce yetkilendirme için **PreAuthorize** anotasyonu kullanılır

```
public class BusinessService {  
  
    @PreAuthorize("hasRole('ROLE_EDITOR')")  
    public void secureMethod() {  
        //...  
    }  
}
```

PreFilter, PostFilter ve PostAuthorize annotasyonları genellikle ACL tarafında ve metot return değerinin yetkilendirilmesinde kullanılır

Secured Annotasyonu ile Metot Erişiminin Yetkilendirilmesi

- ApplicationContext içine `<security:global-method-security secured-annotations="enabled"/>` tanımı eklenir
- Servis metotlarında ise **@Secured** annotasyonu ile roller belirtilir

```
public interface BusinessService {  
    @Secured(value={"ROLE_EDITOR"})  
    public String secureMethod1();  
  
    @Secured(value={"IS_AUTHENTICATED_FULLY"})  
    public String secureMethod2();  
}
```

JSR-250 ile Metot Erişiminin Yetkilendirilmesi

- ApplicationContext içine `<security:global-method-security jsr250-annotations="enabled"/>` tanımı eklenir
- Servis metotlarında **@RolesAllowed** annotasyonu ile roller belirtilir

```
public interface BusinessService {  
    @RolesAllowed(value={"ROLE_EDITOR"})  
    public String secureMethod();  
}
```

- **@PermitAll**, **@DenyAll** gibi annotasyonlarda mevcuttur

İletişim

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

