

Request ve Session Scope Bean Tanımları



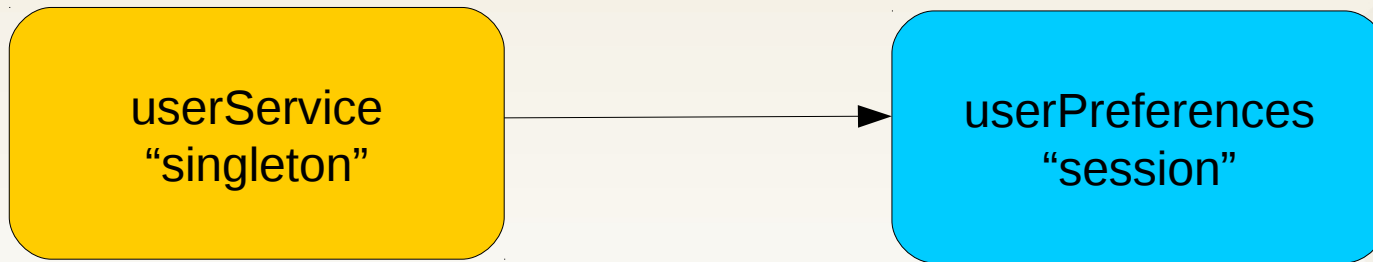
Request ve Session Scope'lar

- Spring Container **HTTP request ve session scope bean'leri** desteklemektedir
- **Request scope** bir bean instance'ının ömrü **web request'idir**
- Her web request'inde yeni bir bean yaratılır
- **Session scope** bir bean instance'ının ömrü **ise kullanıcı oturumu boyunca**dır
- Her farklı kullanıcı oturumu için yeni bir bean yaratılır

Request ve Session Scope'lar

- Request ve session scope'lar sadece **WebApplicationContext** tarafından desteklenir
- **ClassPathXmlApplicationContext** içerisinde kullanılırlarsa **IllegalStateException** fırlatılır
- Implementasyonunda Spring **proxy** örüntüsünden yararlanmıştır

Request/Session Scope'lar ve Scoped Proxy

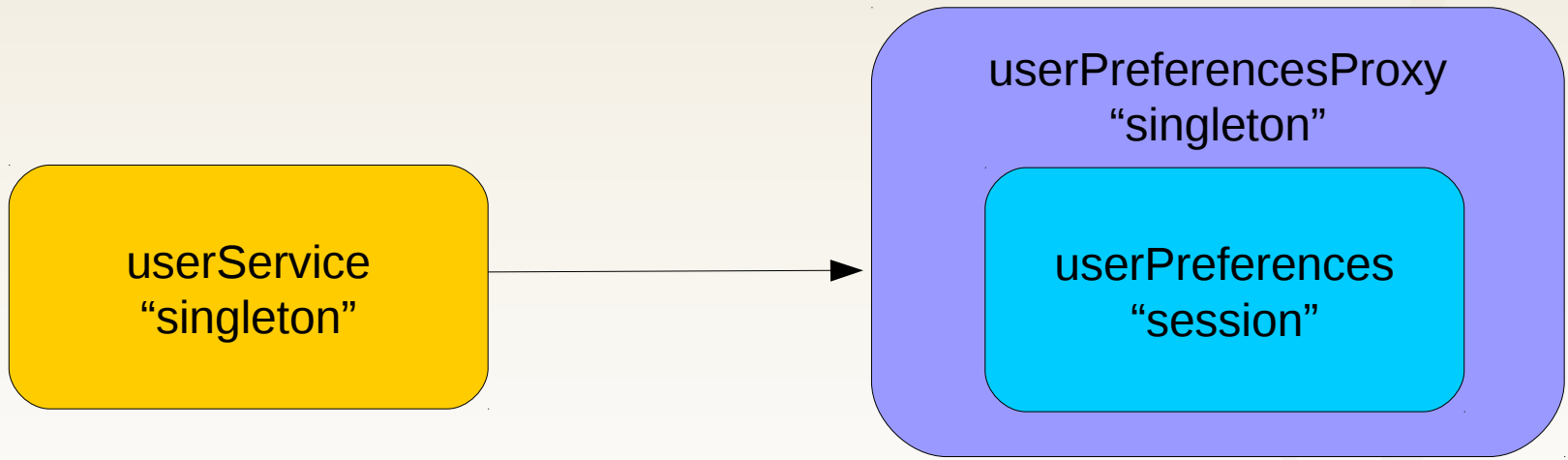


Request/Session Scope'lar ve Scoped Proxy

```
<bean id="userService" class="x.y.UserServiceImpl">  
    <property name="userPreferences"  
ref="userPreferences" />  
</bean>  
  
↓  
<bean id="userPreferences"  
class="x.y.UserPreferencesImpl" scope="session">  
    <aop:scoped-proxy />  
</bean>
```

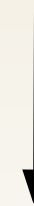
Request veya session scope bir bean'in bağımlılık olarak enjekte edilebilmesi için <aop:scoped-proxy/> tanımı oldukça önemlidir!

Request/Session Scope'lar ve Scoped Proxy



Request/Session Scope'lar ve Scoped Proxy

```
<bean id="userPreferences"
class="x.y.UserPerferencesImpl" scope="session">
    <aop:scoped-proxy proxy-target-
class="false" />
</bean>
```



Default olarak class proxy oluşturulur, eğer **proxy-target-class="false"** ise interface tabanlı proxy yöntemi devreye girer
Bu durumda bean'ın en azından bir interface'i implement etmesi gerekir
Ayrıca bean bağımlılıklarının da interface üzerinden kurulmuş olması gerekir

```
<bean id="userService" class="x.y.UserServiceImpl">
    <property name="userPreferences"
ref="userPreferences" />
</bean>
```

@Scope ve Proxy Oluşturma

Request ve session scope bean'ları tanımlarken proxy modunun belirtilmesi gerekir. **Default proxy modu NO'dur.**

```
@Component
@Scope(value="session", proxyMode=ScopedProxyMode.INTERFACES)
public class UserPreferencesImpl implements UserPreferences {
    // ...
}
```

```
<context:component-scan
    base-package="com.javaegitimleri"
    scoped-proxy="targetClass"/>
```

Component scan sürecinde bu davranış bütün scoped bean'ler için geçerli olacak şekilde de değiştirilebilir

Request ve Session Scope'ların Konfigürasyonu

- Spring Web MVC Framework (**DispatcherServlet**) kullanılırken scoped bean tanımlarının çalışması için **herhangi özel ayara gerek yoktur**
- Eğer istekler Spring Web MVC tarafından ele alınmıyor ise
 - **RequestContextListener** web.xml de tanımlanmalıdır
 - JSF, Struts, Vaadin, GWT, Wicket vb.

Request ve Session Scope'ların Konfigürasyonu

web.xml

```
<web-app...>
```

```
...
```

```
<listener>
```

```
<listener-class>
```

```
org.springframework.web.context.request.RequestContextListener
```

```
</listener-class>
```

```
</listener>
```

```
...
```

```
</web-app>
```



Vaadin, JSF, Struts, ZK gibi UI frameworkleri ile çalışırken scoped bean'ların çalışabilmesi için gereklidir

İletişim

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

