

Hazırlayan: Barış Dere  
www.barisdere.com @BarisDere  
baris.dere@gmail.com

# Spring Framework Eđitimi

Ders 3: Scope kullanımı



[Http://www.youtube.com/barisdere](http://www.youtube.com/barisdere)

# Ders içeriği

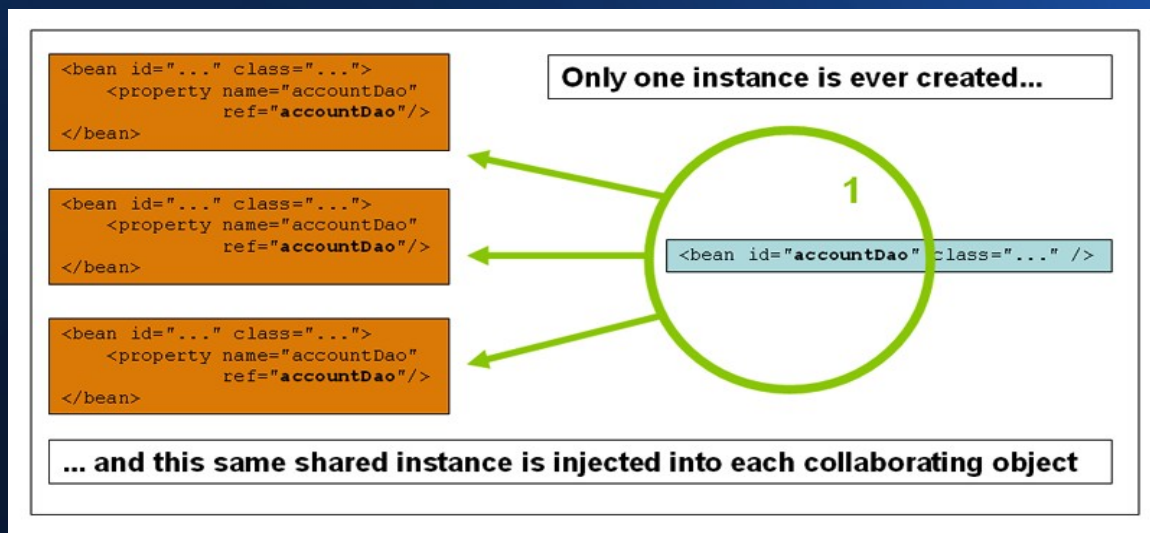
- Scope nedir
- Singleton scope
- Prototype scope
- Web scopes (request, session, globalsession)
- Web scopes kullanımı

# Scope nedir

- Her oluşturulan bean bir scope kullanır
- Kapsam, faaliyet ve kullanım alanını belirtir
- Aynı bean farklı scope kullanarak değişik şekilde hareket eder

# Singleton scope

- Tüm bean ler standart olarak singleton scope kullanırlar
- Bean nesnesi sadece bir kere oluşturulur ve her sorgulamada bu nesne kullanılır



# Singleton scope (2)

- Geleneksel Singleton tasarım kalıbından farklı
- Nesne classloader yerine container içerisinde oluşturulur

# Singleton scope (3)

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

  <bean id="company" class="com.spring.training.Company"></bean>

</beans>
```

```
public static void main(String[] args) {

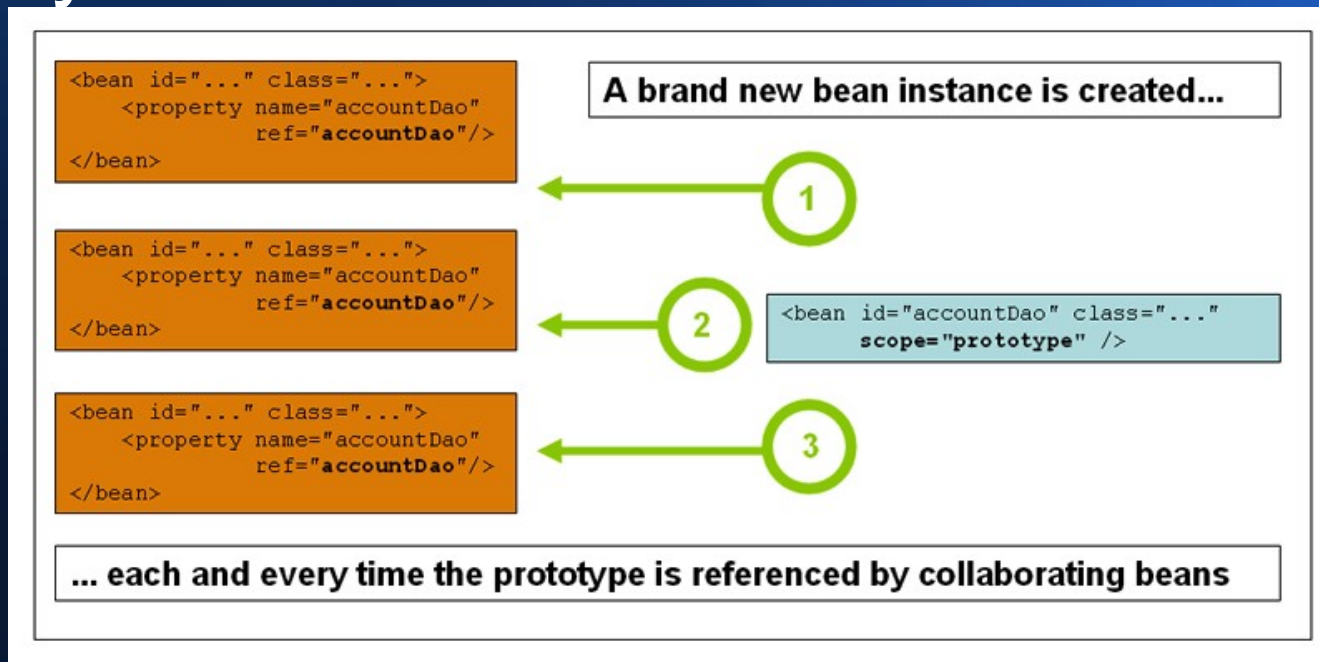
    ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");

    System.out.println(context.getBean("company", Company.class));
    System.out.println(context.getBean("company", Company.class));
    System.out.println(context.getBean("company", Company.class));
    System.out.println(context.getBean("company", Company.class));
}
```

```
com.spring.training.Company@a81b1fb
com.spring.training.Company@a81b1fb
com.spring.training.Company@a81b1fb
com.spring.training.Company@a81b1fb
```

# Prototype scope

- Singleton scope tam tersi
- Bean nesnesi her sorgulamada oluşturulur ve aynı nesne kullanılmaz



# Prototype scope (2)

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

  <bean id="company" class="com.spring.training.Company" scope="prototype"></bean>

</beans>
```

```
public static void main(String[] args) {

  ApplicationContext context = new ClassPathXmlApplicationContext("spring-config.xml");

  System.out.println(context.getBean("company", Company.class));
  System.out.println(context.getBean("company", Company.class));
  System.out.println(context.getBean("company", Company.class));
  System.out.println(context.getBean("company", Company.class));
}
```

```
com.spring.training.Company@580eac5a
com.spring.training.Company@2b5575e0
com.spring.training.Company@77addb59
com.spring.training.Company@371bed71
```



# Prototype scope (3)

- Singleton bir bean içerisinde prototype olan başka bir bean varsa bu prototype singleton olur.
- Bean'ler container başlatılırken oluşturulur
- Method Injection veya kendimiz `new XXX()`

# Web scopes

- Request, Session, Global Session
- Web için kullanılan ApplicationContext sınıflarıyla kullanılabilir (örnek; XmlWebApplicationContext)
- Aksi takdir de IllegalStateException hatası alırsınız

# Web scopes (2)

- Servlet 2.4 ve daha yeni web container için

```
<web-app>
  <listener>
    <listener-class>
      org.springframework.web.context.request.RequestContextListener
    </listener-class>
  </listener>
</web-app>
```

- Servlet 2.3 ve daha eski web container için

```
<filter>
  <filter-name>requestContextFilter</filter-name>
  <filter-class>org.springframework.web.filter.RequestContextFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>requestContextFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

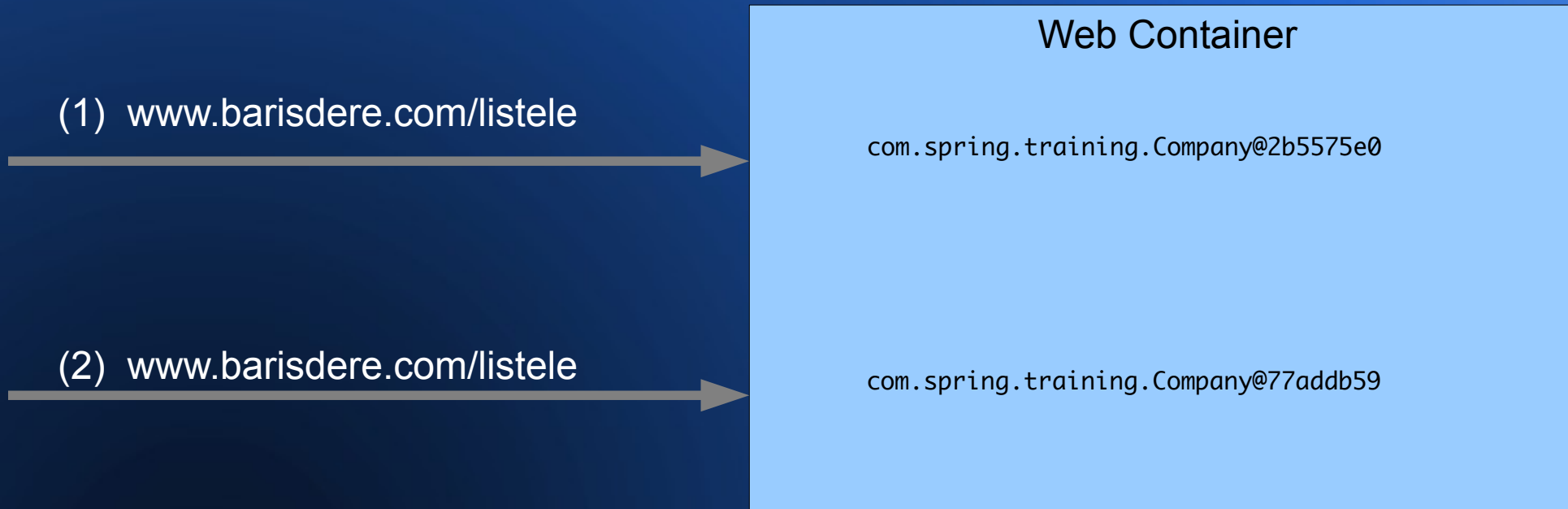
# Request scope

- Her HTTP isteği için yeni bir bean yaratılır
- HTTP isteği sona erince bean yok edilir

```
<beans>  
  <bean id="company" class="com.spring.training.Company" scope="request"/>  
</beans>
```

# Request scope (2)

```
<bean id="company" class="com.spring.training.Company" scope="request" />
```



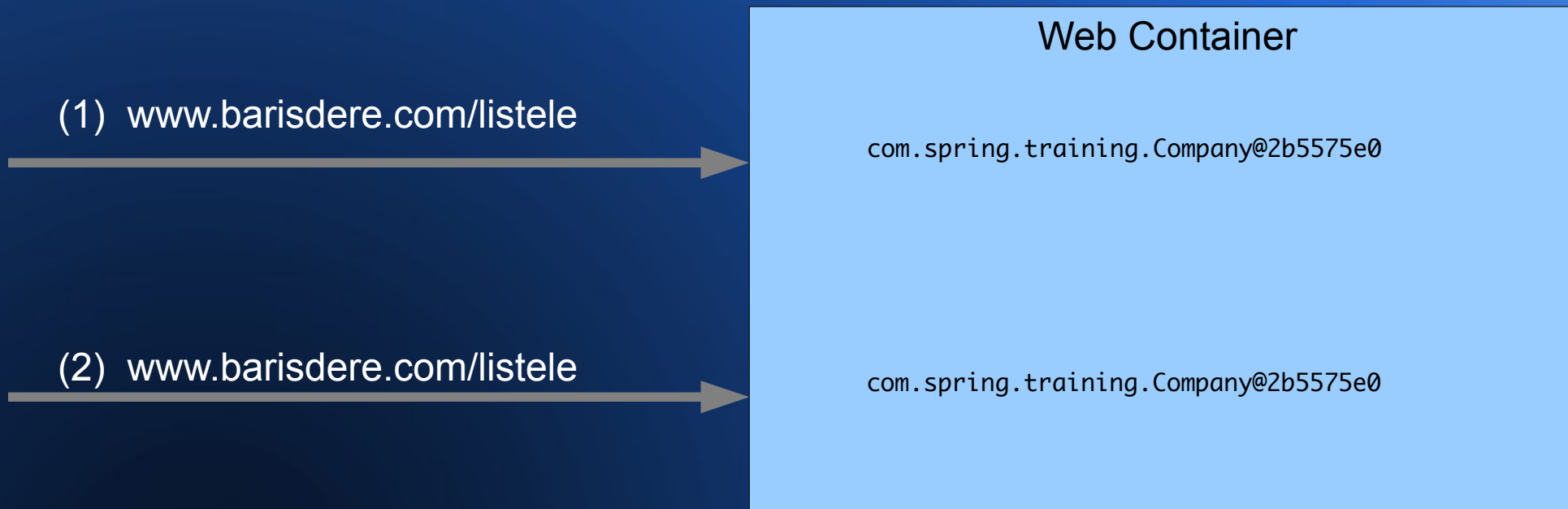
# Session scope

- HTTP session başlamasıyla oluşturulur ve session süresince değişmez.
- HTTP session sona erince bean yok edilir

```
<beans>  
  <bean id="company" class="com.spring.training.Company" scope="session"/>  
</beans>
```

# Session scope (2)

```
<bean id="company" class="com.spring.training.Company" scope="session"/>
```



# Global session scope

- Sadece Portlet uygulamaları için kullanılır
- Portlet'lerin birbirleriyle bilgi paylaşımını sağlıyor
- Web uygulamaları için kullanıldığı takdirde session scope görevi görür

```
<beans>  
  <bean id="company" class="com.spring.training.Company" scope="globalSession"/>  
</beans>
```



# Web scope kullanımı

- Web scope içeren her bean aop scoped proxy kullanmalıdır

```
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:aop="http://www.springframework.org/schema/aop" xsi:schemaLocation="
http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/aop/spring-aop-3.0.xsd">

<bean id="company" class="com.spring.training.Company" scope="request">
    <aop:scoped-proxy />
</bean>

<bean id="customer" class="com.spring.training.Customer">
    <property name="company" ref="company" />
</bean>

</beans>
```

# Ders notları

- Presentasyon dosyalarını download etmek için:  
[www.barisdere.com](http://www.barisdere.com)
- Tüm videolar için:  
<http://www.youtube.com/barisdere>
- Takip etmek için: @BarisDere
- Bana ulaşmak için: [baris.dere@gmail.com](mailto:baris.dere@gmail.com)