

# Spring Bean Scope Tanımları



# Bean Scope Tanımları

- Bean tanımına göre oluşturulacak **nesnelerin ömürlerini** kontrol etmek mümkündür
- Bu işleme “**scoping**” adı verilir
- Spring container tarafından built-in **desteklenen scope'lar**:
  - Singleton, prototype, request, session ve globalSession

# Singleton

- Container'ın ömrü boyunca **tek bir instance** yaratılır
- **Default** scope'dur
- Nesnenin **yaşam döngüsü** Spring Container tarafından yönetilir
- **GoF Singleton** örüntüsünden **farklıdır**
  - GOF'da JVM veya ClassLoader genelinde tek instance vardır

# Prototype

- `ApplicationContext.getBean("name")` çağrısı her seferinde yeni bir instance yaratır
- Nesne yaratıldıktan sonra Spring Container'ın **kontrolünden çıkar**
- Bir açıdan Spring Container, Java'daki **new operatörünün** görevini devralmıştır

# Scope Tanımlarına Örnekler

```
<bean id="foo" class="x.y.Foo" />
```

Default singleton'dur  
Container genelinde bu  
bean  
tanımından tek bir nesne  
oluşturulur

```
<bean id="bar" class="x.y.Bar"  
scope="singleton" />
```

Explicit belirtmeye gerek  
yoktur

```
<bean id="baz" class="x.y.Baz"  
scope="prototype" />
```

Her `getBean("baz")` çağırısı yeni  
bir instance yaratır

# İletişim

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

