

# Bean Scope Lifecycle Callback Metotları



# Lifecycle Callback Metotları

- Spring Container bean'ler **yaratılıp bağımlılıkları enjekte edildikten sonra** bean'lerin bir takım metotlarını invoke edebilir
- Bu metotlara **initialization metotları** denir
- Benzer şekilde bean instance'larının **ömrü sonlandığı vakit** de başka bir takım metotlar invoke edilebilir
- Bunlara da **destruction metotları** denir

# Lifecycle Callback Metotları: Bean Initialization

```
public class Foo {  
    public void init() {  
        // bean ile ilgili initialization islemi  
        yapılır...  
    }  
}
```

Herhangi bir arayüz implement etmeye Gerek yoktur

input argüman almayan herhangi bir metot initialization için kullanılabilir

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

Bean instance yaratıldıktan ve bağımlılıkları enjekte edildikten sonra initialization için bu metot çağrılır

## Metotları: Bean Initialization

```
public class Bar implements InitializingBean {  
    public void afterPropertiesSet() {  
        // bean ile ilgili initialization işlemi  
        yapılır...  
    }  
}
```

Initialization işlemi için diğer alternatif Spring'in  
InitializingBean arayüzünü implement etmektir

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

# Lifecycle Callback Metotları: Bean Initialization

```
public class Foo {  
  
    @PostConstruct  
    public void init() {  
        // init işlemleri...  
    }  
  
}
```

# Lifecycle Callback Metotları: Bean Destroy

**public class Foo** { → Herhangi bir arayüz implement etmeye Gerek yoktur

```
    public void cleanup() {  
        // connection release gibi destruction  
        işlemleri yapılır...  
    }  
}
```

→  
Input argüman almayan herhangi bir metot  
destruction için kullanılabilir

```
<bean id="foo" class="x.y.Foo" destroy-method="cleanup"/>
```

→  
Container kapatılırken veya bean instance  
Scope dışına çıktığında çağrılır  
Prototype bean'lar için anlamlı değildir

# Lifecycle Callback Metotları: Bean Destroy

```
public class Bar implements DisposableBean {  
    public void destroy() {  
        // connection release gibi destruction islemleri  
        yapılır...  
    }  
}
```

↓

Destroy işlemi için diğer alternatif Spring'in DisposableBean Arayüzünü implement etmektir

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

# Lifecycle Callback Metotları: Bean Destroy

```
public class Foo {  
  
    @PreDestroy  
    public void destroy() {  
        // destruction işlemleri...  
    }  
}
```



# @PreDestroy'un Aktivasyonu

- **@PostConstruct** ve **@PreDestroy** JSR-250 annotasyonlarıdır
- **JSR-250**, Java SE için ortak annotationların belirlendiği bir spec'dir
- Spring Container içerisinde aktive olmaları için **<context:annotation-config/>** namespace elemanının tanımlı olması gerekir

# ApplicationContext Shutdown Hook

- Singleton scope bean'lerin **ömürleri** uygulamanın kapanması ile son bulur
- Bu aşamada ApplicationContext kapatılır ve **singleton bean'ler destroy** edilir
- Ancak standalone uygulamalarda **main metodu sonlandığı vakit** ApplicationContext'in bundan haberdar olması gerekir

# ApplicationContext Shutdown Hook

```
public final class StandaloneApp {  
    public static void main(final String[] args) throws  
    Exception {  
        AbstractApplicationContext appContext  
            = new ClassPathXmlApplicationContext(new String []  
            {"beans.xml"});  
    }  
}
```

Standalone uygulamalar içindir

Web uygulamalarında gerek yoktur

```
// JVM'e shutdown hook atılır
```

```
appContext.registerShutdownHook();
```

```
// uygulama ile ilgili diğer ifadeler...
```

```
}
```

JVM main metodundan çıktığı anda container'ın sonlandırılması sağlanır. Burada yapılan şey **Runtime.getRuntime().addShutdownHook()** metodu ile bir Thread nesnesi JVM'e register edilir. JVM sonlandığı vakit daha önceden register Thread'ler çalıştırılarak içerisindeki işlemler gerçekleştirilir. Burada da ApplicationContext close işlemi gerçekleştirilmektedir.

# İletişim

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

