

FactoryBean Kabiliyetleri



FactoryBean ve Bağımlılıklar

- FactoryBean tanımlarındaki bağımlılıklar hedef bean'e değil, FactoryBean instance'ına enjekte edilir
- Bu bağımlılıklar **getObject() metodunda** hedef nesne yaratılırken ona aktarılabilir

```
<bean id="foo"  
class="com.examples.FooFactoryBean">  
  
    <property name="bar" ref="bar"/>  
  
</bean>  
  
<bean id="bar" class="com.examples.Bar"/>
```

```
public class FooFactoryBean implements  
FactoryBean<Foo> {  
  
    private Bar bar;  
  
    public void setBar(Bar bar) {  
        this.bar = bar;  
    }  
  
    @Override  
    public Foo getObject() throws Exception {  
        Foo foo = new Foo();  
        foo.setBar(bar);  
        return foo;  
    }  
    ...  
}
```

FactoryBean ve Autowire

- Bean autowire işlemi **FactoryBean'e uygulanır**, getObject() metodu ile dönen hedef bean'a uygulanmaz

```
<bean id="foo"  
class="com.javaegitimleri.petclinic.service.FooFactoryBean"  
autowire="byType"/>
```

```
<bean id="bar"  
class="com.javaegitimleri.petclinic.service.Bar"/>
```

FactoryBean ve Lazy Davranış

- Spring FactoryBean yöntemi ile yaratılan bean'ler default eager davranışın aksine **lazy olarak** yaratılırlar
- Başka bir ifade ile **FactoryBean.getObject()** metodu bean'e ne zaman ihtiyaç duyulur ise o vakit invoke edilecektir
- Eğer bootstrap aşamasında target bean'in eager biçimde yaratılması istenirse **SmartFactoryBean** arayüzü implement edilmelidir

SmartFactoryBean

```
public class FooFactoryBean implements SmartFactoryBean<Foo> {

    @Override
    public Foo getObject() throws Exception {
        return new Foo();
    }

    @Override
    public boolean isEagerInit() {
        return true;
    }

    @Override
    public Class<?> getObjectType() {
        return Foo.class;
    }

    ...
}
```

FactoryBean'e Erişmek

```
Foo foo = (Foo) applicationContext.getBean("foo");
```

```
FooFactoryBean fooFactoryBean =  
    (FooFactoryBean) applicationContext.getBean("&foo");
```

Bean ismi ile erişim sonucu getObject() metodu ile yaratılan nesne dönülecektir. Eğer bean isminin başına "&" konulursa bu durumda FactoryBean'e erişilebilir

```
Foo foo = applicationContext.getBean(Foo.class);
```

```
FooFactoryBean fooFactoryBean =  
    applicationContext.getBean(FooFactoryBean.class);
```

Eğer ApplicationContext'de bir sınıftan tek bir bean mevcut ise bu durumda sınıf ismi ile de hedef bean'e veya FactoryBean'e erişmek mümkündür

İletişim

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

