XML Tabanlı Container Konfigürasyonu

harezmi bilişim çözümleri

XML Tabanlı Konfigürasyon Nasıl Yapılır?

- XML konfigürasyon dosyaları içerisinde **bean tanımları** `<beans></beans>` root elemanı altında gerçekleştirilir
- Her bean için `<bean></bean>` elemanı ile **ayrı bir bean tanımı** yapılır
- Bean'in **ihtiyaç duyduğu bağımlılıklar** da `<property></property>` elemanları ile enjekte edilir

XML Tabanlı Konfigürasyon Nasıl Yapılır?

- Genellikle her **bean tanımı** uygulamadaki **bir Java nesnesi**ne karşılık gelir
- **Property tanımları** ile yapılan da, o nesnenin bağımlılıklarının **enjekte edilmesi**dir
- Bean tanımları **bir veya birkaç XML konfigürasyon dosyası**na yayılabilir

XML Tabanlı Konfigürasyon Nasıl Yapılır?

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...>

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

  <bean id="foo" class="x.y.Foo">
      <property name="myBar" ref="bar"/>
      <property name="age" value="15"/>
  </bean>
  ...

</beans>
```

```
Bar bar = new Bar();
```

```
Foo foo = new Foo();
foo.setMyBar(bar);
foo.setAge(15);
```

Bean İsimlendirme

- Her bean tanımı bir veya daha fazla **identifier**'a (bean ismi) sahiptir
- XML'de **id** veya **name** attribute'ları kullanılarak tanımlanırlar

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

<bean name="bar,myBar" class="x.y.z.Bar"/>

<bean name="/baz" class="x.y.z.Baz"/>
```

Bean İsimlendirme

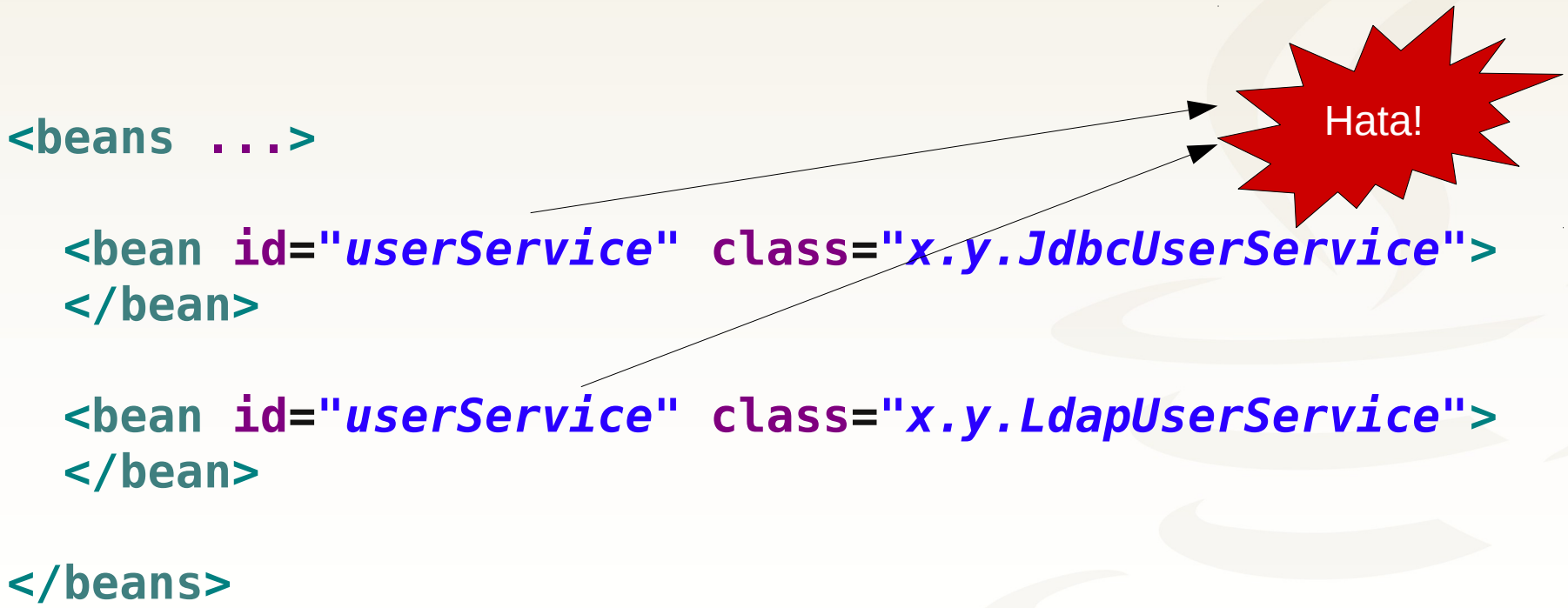
- Bean tanımlarına isim verilmesi **zorunlu değil**dir
- Eğer verilmez ise container **internal** bir isim verir
- Id attribute **tek bir bean ismi** tanımlamaya izin verir
- Bean tanımına **birden fazla isim** vermek için **name** attribute kullanılır
  - Birden fazla isim verilebilir
  - Virgül, boşluk veya noktalı virgül ile ayrılırlar

Bean İsimlendirme

- Aynı XML bean konfigürasyon dosyasında **aynı isimde birden fazla bean** tanımı olamaz
- Ancak **farklı XML dosyalarında** aynı isimde birden fazla bean tanımlanabilir
- Böyle bir durumda ApplicationContext yaratım esnasında sıralamada **sonra gelen** XML dosyasındaki bean tanımı **önce gelen tanımı ezer**
- Sonuç olarak **çalışma zamanında o isimde tek bir bean tanımı** olacaktır

Bean Tanımlarının Ezilmesi (Bean Definition Override)

- Aynı XML konfigürasyon dosyası içerisinde aynı isimde birden fazla bean tanımı olamaz



Bean Tanımlarının Ezilmesi (Bean Definition Override)

- Farklı XML dosyalarında aynı isimli bean tanımı olabilir
- Bu durumda XML dosyalarının yüklenme sırası önem kazanır

beans-service.xml

```
<beans ...>
  <bean id="userService"
    class="x.y.JdbcUserService"/>
</beans>
```

beansOverride.xml

```
<beans ...>
  <bean id="userService"
    class="x.y.LdapUserService"/>
</beans>
```

```
ApplicationContext context =
    new ClassPathXmlApplicationContext(
        "beans-dao.xml",
        "beans-service.xml",
        "beansOverride.xml");
```

XML Import

- XML konfigürasyon dosyalarını başka bir XML konfigürasyon dosya içerisinde **import ederek** de **yükletmek** mümkündür

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...>

  <import resource="beans-dao.xml"/>
  <import resource="classpath:/beans-config.xml"/>

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

İletişim

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