

# XML Tabanlı Konfigürasyonda Collection Tanımları



# Collection Tanımları

- Aşağıdaki XML elemanları ile bean'lar içindeki **Array**, **List**, **Set**, **Map** ve **Properties** tipindeki property değerleri tanımlanabilir
  - `<list/>`
  - `<set/>`
  - `<map/>`

# Collection Tanımlarına Örnekler: List

java.util.List tanımı

```
<bean id="foo" class="x.y.Foo">  
  <property name="someList">  
    <list>  
      <value>primitif deger 1</value>  
      <value>primitif deger 2</value>  
    </list>  
  </property>
```

```
<property name="someAnotherList">  
  <list>  
    <ref bean="myBean1" />  
    <ref bean="myBean2" />  
  </list>  
</property>
```

```
</bean>
```

# Collection Tanımlarına Örnekler: Array

```
<bean id="foo" class="x.y.Foo">  
  <property name="someArray">  
    <list>  
      <value>primitif deger 1</value>  
      <value>primitif deger 2</value>  
    </list>  
  </property>
```

```
<property name="someAnotherArray">  
  <list>  
    <ref bean="myBean1" />  
    <ref bean="myBean2" />  
  </list>  
</property>
```

```
</bean>
```

Array tipindeki bir property de list elemanı ile enjekte edilmektedir

# Collection Tanımlarına Örnekler: Set

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

java.util.Set tanımı

```
<property name="someSet">
```

```
<set>
```

```
<value>herhangi bir String ifade</value>
```

```
<value>herhangi başka bir String ifade</value>
```

```
</set>
```

```
</property>
```

```
<property name="someAnotherSet">
```

```
<set>
```

```
<ref bean="myBean1" />
```

```
<ref bean="myBean2" />
```

```
</set>
```

```
</property>
```

```
</bean>
```

# Collection Tanımlarına Örnekler: Map

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

```
<property name="someMap">
```

```
<map>
```

```
<entry key="key1" value="value1"/>
```

```
<entry key="key2" value="value2"/>
```

```
</map>
```

```
</property>
```

java.util.Map tanımı



```
<property name="someAnotherMap">
```

```
<map>
```

```
<entry key-ref="keyBean1" value-ref="myBean1"/>
```

```
<entry key-ref="keyBean2" value-ref="myBean2"/>
```

```
</map>
```

```
</property>
```

```
</bean>
```

# Collection Tanımlarına Örnekler: Properties

```
<bean id="foo" class="x.y.Foo">  
  <property name="keyValuePairs">  
    <props>  
      <prop key="key1">value1</prop>  
      <prop key="key2">value2</prop>  
    </props>  
  </property>  
</bean>
```

java.util.Properties tanımı

# Java Generics ve Collection Tanımları

```
public class Foo {  
  
    private Map<String, Float> floatValues;  
  
    public void setFloatValues(Map<String, Float> values) {  
        this.floatValues = values;  
    }  
}
```

```
<bean id="foo" class="x.y.Foo">  
    <property name="floatValues">
```

```
        <map>
```

```
            <entry key="key1" value="9.99"/>
```

```
            <entry key="key2" value="2.75"/>
```

```
            <entry key="key3" value="3.99"/>
```

```
        </map>
```

```
    </property>
```

```
</bean>
```

Java generics bilgisi kullanılarak  
değerlerin uygun tipe çevrimi de  
gerçekleştirilir



# Collection Merge Kabiliyeti

```
<bean id="parent" abstract="true">
  <property name="keyValuePairs">
    <props>
      <prop key="key1">value1</prop>
      <prop key="key2">value2</prop>
    </props>
  </property>
</bean>
```

Child bean tanımındaki  
Collection değerleri ile  
parent bean tanımındaki  
collection değerleri  
birleştirilir

```
<bean id="child" class="x.y.Foo" parent="parent">
  <property name="keyValuePairs">
    <props merge="true">
      <prop key="key3">value3</prop>
      <prop key="key2">value2-override</prop>
    </props>
  </property>
</bean>
```

Parent tanımındaki değerlerin sırası child tanımındakilerden önce gelir  
Farklı collection tipleri merge edilemez

# İletişim

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

