

O/X Mapping ve XML Marshal/Unmarshal İşlemleri



Spring O/X Desteği

- Spring Framework içerisinde **XML – nesne dönüşümü** yapmayı sağlayan bir modül mevcuttur
- Java object serialization'a benzetilebilir
- Nesnenin XML'e dönüştürülmesi işlemine **marshal**
- XML'in de tekrar nesneye dönüştürülmesine **unmarshal** adı verilir
- Bu işlemleri yapan bean'lara da **marshaller** ve **unmarshaller** adı verilir

Spring O/X Desteği

- Spring **farklı O/X teknolojilerini** desteklemektedir
- O/X kütüphanesinin Spring ile konfigürasyonu daha kolay olmaktadır
- Spring **Marshaller ve Unmarshallers arayüzleri** üzerinden çalışmayı sağlamaktadır
- **Ortak bir exception hiyerarşisi** sunmaktadır

Marshaller & Unmarshaller

```
public interface Marsheller {
```

Nesneyi XML gösterimine dönüştürür ve Result içerisine yazar

```
void marshal(Object graph, javax.xml.transform.Result result)
    throws XmlMappingException, IOException;
```

```
}
```

Result arayüzünün farklı implemantasyonları mevcuttur. Her bir implemantasyon da farklı XML gösterimlerini kendi içerisinde wrap eder.
DOMResult:org.w3c.dom.Node
SAXResult:org.xml.sax.ContentHandler
StreamResult:java.io.File, java.io.OutputStream, or java.io.Writer

```
public interface Unmarshaller {
```

Source ile verilen XML gösterimi nesneye dönüştürür ve object olarak döner

```
Object unmarshal(javax.xml.transform.Source source)
    throws XmlMappingException, IOException;
```

```
}
```

Source arayüzünün farklı implemantasyonları mevcuttur. Her bir implemantasyon da farklı XML gösterimlerini kendi içerisinde wrap eder.
DOMSource:org.w3c.dom.Node
SAXSource:org.xml.sax.InputSource
StreamSource:org.xml.sax.XMLReader java.io.File, java.io.InputStream, or java.io.Reader

OXM Namespace

- Marshaller konfigürasyonunu kolaylaştırmak için Spring, **OXM namespace**'ini sunar

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:oxm="http://www.springframework.org/schema/oxm"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/oxm
http://www.springframework.org/schema/oxm/spring-oxm-4.1.xsd">

</beans>
```

OXM Namespace

- OXM namespace tarafından **desteklenen** marshaller elemanları şunlardır:
 - jaxb2-marshaller
 - xmlbeans-marshaller
 - castor-marshaller
 - jibx-marshaller

- **JAXB binding compiler**, XSD tanımından Java sınıfları üretmeyi sağlar
- Spring'de **JAXB 2.0 API**'si üzerinden marshalling kabiliyeti sunmaktadır
- **Jaxb2Marshaller** sınıfı hem Marshaller, hem de Unmarshaller arayüzüne sahiptir

```
<oxm:jaxb2-marshaller id="marshaller"  
    contextPath="com.javaegitimleri.petclinic.ws.model" />
```



JAXB compiler tarafından üretilmiş Java sınıflarını içeren paket isimleri
“.” ile ayrılarak belirtilir

```
<oxm:jaxb2-marshaller id="marshaller">  
    <oxm:class-to-be-bound  
        name="com.javaegitimleri.petclinic.ws.model.HelloWorldRequest" />  
    <oxm:class-to-be-bound  
        name="com.javaegitimleri.petclinic.ws.model.HelloWorldResponse" />  
</oxm:jaxb2-marshaller>
```



contextPath yerine diğer bir alternatif de teker teker sınıfları listelemektir

Castor

- Castor açık kaynak kodlu bir XML binding framework'tür
- **CastorMarshaller** sınıfı hem Marshaller, hem de Unmarshaller arayüzüne sahiptir

```
<oxm:castor-marshaller
    id="marshaller"
    mapping-location="classpath:/castor/mapping.xml" />
```

XML Beans

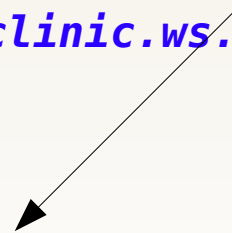
- XMLBeans'de bir XML binding aracıdır
- Diğer araçlardan farklı olarak üretilen Java sınıfları **XmlObject** sınıfından türerler ve **binding bilgisini** kendi içlerinde barındırırlar
- **XmlBeansMarshaller** sınıfı hem Marshaller, hem de Unmarshaller arayüzüne sahiptir

```
<oxm:xmlbeans-marshaller id="marshaller"/>
```

- Binding definition **binding kuralları** içerir
- Bu kurallar Java nesnelerinin XML dönüşümünün **nasıl yapılacağını** tanımlar
- JiBX binding compiler **class dosyalarını enhance** ederek binding işlemi için ilave kodlar ekler
- **JibxMarshaller** sınıfı hem Marshaller, hem de Unmarshaller arayüzüne sahiptir

```
<oxm:jibx-marshaller id="marshaller1" target-  
class="com.javaegitimleri.petclinic.ws.model.HelloWorldRe  
quest"/>
```

```
<oxm:jibx-marshaller id="marshaller2" target-  
class="com.javaegitimleri.petclinic.ws.model.HelloWorldRe  
sponse"/>
```



Dönüşümü yapılacak Java nesnesinin sınıfı target-class attribute'u ile belirtilir. Sadece bir sınıf belirtilebilir. Her bir sınıf için ayrı ayrı marshaller tanımlamak gerekir

XStream

- Java nesnelerinin **XML'e serialize** ve **XML'den deserialize** edilmesini sağlayan bir kütüphanedir
- XStream bir **serialization aracıdır**, XML **binding kabiliyeti yoktur**
- Dolayısı ile XML **web servislerinde** kullanımı çok da **uygun değildir**

XStream

```
<bean id="xstreamMarshaller"
class="org.springframework.xml.xstream.XStreamMarshaller">
  <property name="supportedClasses">
    <list>
      <value>
        com.javaegitimleri.petclinic.ws.model.HelloWorldRequest
      </value>
      <value>
        com.javaegitimleri.petclinic.ws.model.HelloWorldResponse
      </value>
    </list>
  </property>
</bean>
```

Xstream herhangi bir Java nesnesinin XML serialization'ını veya deserialization'ını yapabilir. Web üzerinden indirilen XML verisinin deserialization'ı güvenlik açığına yol açabilir. Bu nedenle Xstream'in sadece spesifik sınıflar için deserialization yapmasını sağlamak için **supportedClasses** attribute'u kullanılır

İletişim

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

