

JSP Custom Tags



Custom JSP Tags

- JSP sayfalarını oluşturmak için custom **HTML veya XML benzeri tag**'ler tanımlanabilir
- Bu tag'ler arka planda **spesifik reusable bir kod parçacığına** karşılık gelmektedir
- JSP Container custom **bir JSP tag**'i gördüğünde arkada buna karşılık gelen **Java kodunu** çalıştırır
- Tag'ler bir araya getirilerek **tag kütüphaneleri** oluşturulur

Custom JSP Tags

- Tag'ler JSP içeriği şeklinde **tag dosyaları** içerisinde yazılır
- Java kodu ile de tag yazmak mümkündür
- Tag dosyaları da projenin **WEB-INF/tags** isimli dizine kopyalanmalıdır
- Ya da tag dosyaları ve tag kodları **JAR dosyası** içerisinde **/META-INF/tags** altına konabilirler
- Bu şekilde tag'ler farklı projelerde **yeniden kullanılabilirler**

Tag Dosyası ile Custom Tag Tanımlama

/WEB-INF/tags/greeting.tag

```
<%@ attribute name="message" required="true"%>
<%@ attribute name="name" required="true"%>
<h2>
    <font color="black">${message}, ${name}!</font>
</h2>
```

Tag tanımı

index.jsp

```
<%@ taglib prefix="g" tagdir="/WEB-INF/tags"%>

<html>
    <body>
        <g:greeting message="Hello" name="Kenan" />
    </body>
</html>
```

Tag kullanımı

JSP container dosya ismini aynı zamanda tag ismi olarak da kabul eder

Custom tag'lerin kullanılacağı JSP sayfalarında önce taglib direktifi ile tanım yapmak gerekir

Java ile Custom Tag Örneği

```
public class SendMessageTag extends SimpleTagSupport {
```

```
    private String from;  
    private String to;
```

```
    public void setFrom(final String from) {  
        this.from = from;  
    }
```

```
    public void setTo(final String to) {  
        this.to = to;  
    }
```

Attribute'ların değerlerinin elde edilebilmesi için setter metotlar tanımlanmalıdır

```
@Override
```

```
public void doTag() throws JspException, IOException {
```

```
    final StringWriter writer = new StringWriter();  
    getJspBody().invoke(writer);
```

Tag'in body'sindeki girilmiş değeri döner

```
    String message = writer.toString();
```

PageContext nesnesine erişim sağlar

```
    getJspContext().getOut().println("Message from '"  
        + from + "' to '" + to + "'. Message content : " + message);
```

```
}
```

```
}
```

Tag Library Descriptors

- Tag dosyalarını farklı projelerde kullanmak üzere **jar arşivi içerisinde** toplamak
- Yada **Java kodu ile yazılan tag'leri** kullanabilmek için **TLD** tanımlamak gerekir
- TLD bir **XML** dosyadır, uzantısı ***.tld** olmalıdır
- Tag kütüphanesindeki **her bir tag elemanı** bu dosyada yer alır
- Projede **/WEB-INF/** dizini altında veya jar dosyası içerisinde **/META-INF/** dizini altında olabilirler

TLD Örneği

```
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>2.0</jsp-version>
  <short-name>PetClinic TLD</short-name>
  <uri>http://www.java-egitimleri.com/petclinic/tags</uri>

  <tag>
    <name>greeting</name>
    <tag-file>/WEB-INF/tags/greeting.tag</tag-file>
  </tag>

  <tag>
    <name>send-message</name>
    <tag-class>com.javaegitimleri.petclinic.web.SendMessageTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
      <name>from</name>
      <required>true</required>
    </attribute>
    <attribute>
      <name>to</name>
      <required>true</required>
    </attribute>
  </tag>
</taglib>
```

Java Standart Tag Library (JSTL)

- **JSTL** kütüphanesi pek çok JSP uygulamasında **sıklıkla ihtiyaç duyulan ortak tag'leri** bir araya getirmektedir
- Projeye ayrı bir **JAR** olarak eklenmelidir
- JSTL bünyesinde **değişik tag kütüphaneleri** mevcuttur
 - **Core** : En sık kullanılan, if, for, when, out, redirect gibi fonksiyonları içerir
 - **Format** : Text, date, number gibi çıktıları formatlama ile ilgili fonksiyonları içerir
 - **SQL** : DataSource oluşturma, SQL işlemleri ile ilgili fonksiyonları içerir
 - **XML** : XML processing ile ilgili fonksiyonları içerir
 - **JSTL Functions** : Çoğunlukla String işlemler ile ilgili fonksiyonları içerir

JSTL Taglib Kullanımı

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<html>
  <body>
    <sql:setDataSource var="ds" driver="org.h2.Driver"
      url="jdbc:h2:tcp://localhost/test" user="sa" password="" />
    <sql:query dataSource="${ds}" var="result">
      SELECT * from Employees;
    </sql:query>
    <table width="100%">
      <tr>
        <th>Emp ID</th><th>First Name</th><th>Last Name</th><th>Age</th>
      </tr>
      <c:forEach var="row" items="${result.rows}">
        <tr>
          <td><c:out value="${row.id}" /></td>
          <td><c:out value="${row.first}" /></td>
          <td><c:out value="${row.last}" /></td>
          <td><c:out value="${row.age}" /></td>
        </tr>
      </c:forEach>
    </table>
  </body>
</html>
```

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)