

Üst Düzey Programlama

JSP Etiketletleri(Tags)

Kişiye Özel JSP Etiketleri

JSP Etiketleri, JSP sayfasında yazılan Java script leri yerine kullanılabilen HTML etiketleri yapısına uygun JSP etiketleridir.

Kişiye özel olarak geliştirilebileceği gibi Sun, Apache gibi firmaların geliştirdikleri hazır etiket kütüphaneleri de kullanılabilir.

`<jsp:useBean`
`<jsp:setProperty ...` gibi jsp etiketleridir.

Öncelikle kendimize özel etiketlerin nasıl oluşturulduğunu ve nasıl kullanıldığını görelim.

Kişiye Özel JSP Etiketleri

JSP Etiketlerini(Tag) kullanabilmek için 3 ayrı bileşene ihtiyaç vardır:

1. Etiket işleyici(Tag Handler) sınıf
2. Etiket(Tag) kütüphanesini tanımlayan bir dosya. Bu dosya xml biçimlidir. Kullanılan etiketlere karşılık olarak kullanılacak sınıflar tanımlanır.
3. Etiket kütüphanesini kullanan JSP dosyası.

Etiket İşleyici (Tag Handler) Sınıfı

JSP sayfasında sizin kullandığınız etiket ile karşılaşıldığında gerçekleştirilmesi gerekli olan gerçek kodun yazıldığı sınıftır.

Bu sınıf `javax.servlet.jsp.tagext.Tag` arayüzünü gerçekleştirmelidir.

Bu arayüzü gerçekleştirmek için, hazırlanmış olan `TagSupport` ya da `BodyTagSupport` isimli iki sınıftan bir tanesinden sınıf türetmeliyiz.

Etiket İşleyici (Tag Handler) Sınıfı

```
OrnekTag.java x
package ce.ders06;

import java.io.IOException;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.BodyContent;
import javax.servlet.jsp.tagext.BodyTagSupport;

public class OrnekTag extends BodyTagSupport {

    public int doStartTag() throws JspException, JspException {
        try {
            JspWriter out = pageContext.getOut();
            out.print("Özel Etiket Örneği");
        } catch (IOException e) {
            System.out.println("Hata:" + e);
        }
        return (SKIP_BODY);
    }
}
```

Etiket Kütüphanesi Tanımlayıcısı (Tag Library Descriptor)

İkinci olarak bu etiketi sunucuya tanıtan bir tanımlayıcı dosya yazılmalıdır. Bu dosya XML biçimlidir.

```
my_etiketlerim.tld x
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
  <tlib-version>1.0</tlib-version>
  <short-name>my_etiketlerim</short-name>
  <uri>/WEB-INF/tlds/my_etiketlerim</uri>

  <tag>
    <name>OrnekTag</name>
    <tag-class>ce.ders06.OrnekTag</tag-class>
    <body-content>JSP</body-content>
  </tag>
</taglib>
```

JSP de Etiketin Kullanımı

Yazmış olduğumuz bu etiketi JSP lerde kullanabiliriz.
Bunun için;

`<%@ taglib uri="..." prefix="..." %>` ifadesi sayfada yazılmalıdır.

uri ifadesi JSP den tanımladığımız etiket kütüphanesini kullanmak için tanımlanır.

prefix ifadesi, kendi oluşturduğunuz etiketlerin diğer etikerlerden ayırt edilmesi için kullanılır.

JSP de Etiketin Kullanımı

```
Ornek01.jsp x
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

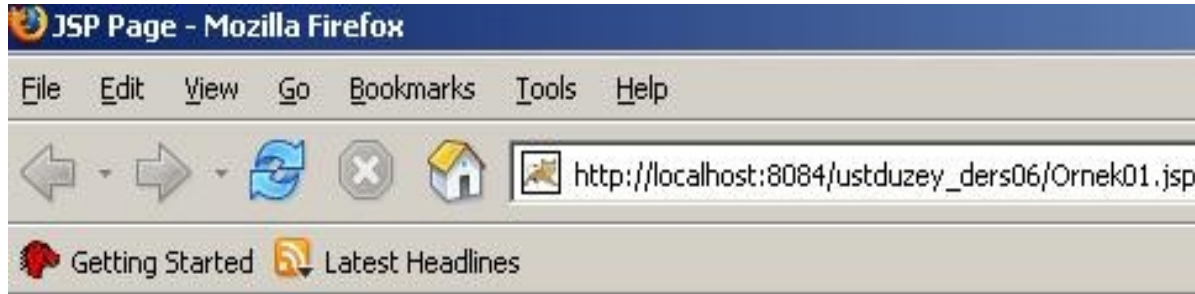
<%@taglib uri="/WEB-INF/tlds/my_etiketlerim.tld" prefix="my_e"%>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
</head>
<body>

    <h1>JSP Etiket Örneği</h1>
    <my_e:OrnekTag>
        asdasd
    </my_e:OrnekTag>

</body>
</html>
```


JSP de Etiketin Kullanımı



JSP Etiket Örneği

Özel Etiket Örneği

Etiketlerde Özellik(Attribute) Kullanımı

`<prefix:ad ozellik1="deger1" ozellik2="deger2".../>`

şeklinde etiketler tanımlamak için kullanılır.

Örneğin, özellik1 e bir değer atadığımızda, etikete karşılık gelen etiket işleyici sınıfına ait olan setOzellik1 isimli metod çalıştırılır.

Bu metodu sınıfa bizim eklememiz gereklidir. Değer olarak sadece String veri tipi kullanılabilir.

```
public void setOzellik1(String deger1){  
    //gerekli olan işler  
}
```

Etiketlerde Özellik(Attribute) Kullanımı

```
MesajTag.java x
import java.io.IOException;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.BodyContent;
import javax.servlet.jsp.tagext.BodyTagSupport;

public class MesajTag extends BodyTagSupport {
    private String mesaj;

    public void setMesaj(String mesaj){
        this.mesaj = mesaj;
    }

    public int doStartTag() throws JspException{
        try {
            JspWriter out = pageContext.getOut();
            out.print("Mesajınız:"+mesaj);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return (SKIP_BODY);
    }
}
```

Etiketlerde Özellik(Attribute) Kullanımı

```
my_etiketlerim.tld * x
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
  <tlib-version>1.0</tlib-version>
  <short-name>my_etiketlerim</short-name>
  <uri>/WEB-INF/tlds/my_etiketlerim</uri>
  <tag>
    <name>OrnekTag</name>
    <tag-class>ce.ders06.OrnekTag</tag-class>
    <body-content>JSP</body-content>
  </tag>
  <tag>
    <name>MesajTag</name>
    <tag-class>ce.ders06.MesajTag</tag-class>
    <body-content>JSP</body-content>
    <attribute>
      <name>mesaj</name>
      <required>>false</required>
    </attribute>
  </tag>
</taglib>
```

Etiketlerde Özellik(Attribute) Kullanımı

Ornek02.jsp x

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

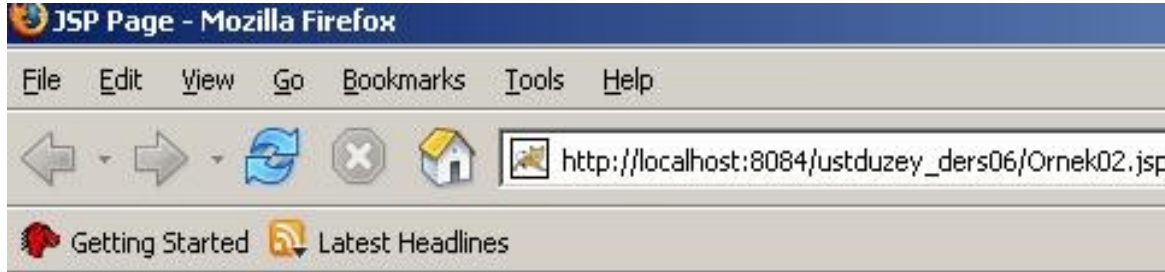
<%@ taglib uri="/WEB-INF/tlds/my_etiketlerim.tld" prefix="my_e" %>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
</head>
<body>

    <h1>JSP Etiket Örneği 2</h1>
    <my_e:MesajTag mesaj="Bilgisayar Mühendisliği"/>

</body>
</html>
```

Etiketlerde Özellik(Attribute) Kullanımı



JSP Etiket Örneği 2

Mesajınız.Bilgisayar Mühendisliği

JSP Etiketleri

`doStartTag()` metodu etiket ile karşılaşıldığında çalışır.
`doEndTag()` metodu etiket sonlandırıldığında çalışır.

Sizin tanımladığınız etiketlerin başlangıç ve bitişi arasındaki bölüme etiket gövdesi (Body) denilir.

`doStartTag()` metodundan geriye (`SKIP_BODY`) gönderilirse gövde işlenmez.

Gövdeninde değerlendirilmesini istiyorsanız, (`EVAL_BODY_INCLUDE`) değeri geriye göndermelisiniz.

`doEndTag()` metodundan sonra sayfanın kalanının yorumlanmasını isterseniz (`EVAL_PAGE`), istemezseniz (`SKIP_PAGE`) değerlerinin çevirmelisiniz.

```
package ce.ders06;
```

```
+ import ...
```

```
public class GovdeEtiket extends BodyTagSupport {
```

```
    private int tekrar;
```

```
    public void setTekrar(String tekrars){
```

```
        try {
```

```
            this.tekrar = Integer.parseInt(tekrars);
```

```
        } catch (Exception e) {
```

```
            this.tekrar = 1;
```

```
        }
```

```
    }
```

```
    public int doAfterBody() throws JspException {
```

```
        if (tekrar-->=1){
```

```
            BodyContent body = getBodyContent();
```

```
            try {
```

```
                JspWriter out = body.getEnclosingWriter();
```

```
                out.println(body.getString());
```

```
                out.print("<br>");
```

```
                body.clearBody();
```

```
            } catch (Exception e) {
```

```
                System.out.println("TekrarEden Tag da hata:"+e);
```

```
            }
```

```
            return (EVAL_BODY_AGAIN);
```

```
        }else{
```

```
            return (SKIP_BODY);
```

```
        }
```

```
    }
```

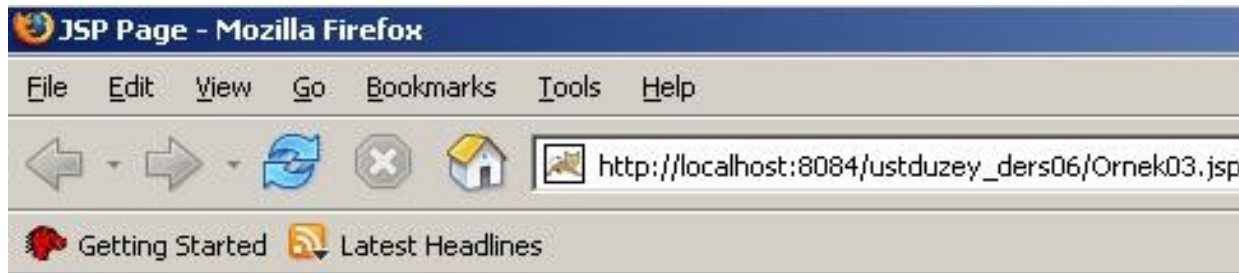


```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.0" xmlns="http://java.sun.com/xml/ns/j2ee"
  <tlib-version>1.0</tlib-version>
  <short-name>my_etiketlerim</short-name>
  <uri>/WEB-INF/tlds/my_etiketlerim</uri>
  <tag>
    <name>OrnekTag</name>
    <tag-class>ce.ders06.OrnekTag</tag-class>
    <body-content>JSP</body-content>
  </tag>
  <tag>
    <name>MesajTag</name>
    <tag-class>ce.ders06.MesajTag</tag-class>
    <body-content>JSP</body-content>
    <attribute>
      <name>mesaj</name>
      <required>>false</required>
    </attribute>
  </tag>
  <tag>
    <name>GovdeEtiket</name>
    <tag-class>ce.ders06.GovdeEtiket</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
      <name>tekrar</name>
      <required>>true</required>
    </attribute>
  </tag>
</taglib>
```

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@ taglib uri="/WEB-INF/tlds/my_etiketlerim" prefix="my_e" %>
```

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <my_e:GovdeEtiket tekrar="10" >
      deee
    </my_e:GovdeEtiket>
  </body>
</html>
```



deee
deee
deee
deee
deee
deee
deee
deee
deee
deee

JSP İfade Dili (Expression Language)

JSP ifade dili, JSP sayfasında bulunan bean ve nesnelerin özelliklerine kolay bir yazım şekli ile ulaşılmasını sağlar.

`${ifade}`

şeklinde yazılır.

Basit karşılaştırma ve matematiksel işlemler yapılabilir.

Temel kullanım amacı, bean lerin özelliklerine hızlı bir şekilde erişmektir.

JSP İfade Dili (Expression Language)

```
ELOrnek05.jsp x
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP İfade Dili(Expression Language)</title>
</head>
<body>
    ${1 > (4/2)}<br>
    ${4.0 > 3}<br>
    ${ (6*5) + 5}<br>
    ${param.kullanici}<br>
    <% session.setAttribute("adi",new String("DENEME")); %>

    session degeri = ${sessionScope.adi}<br>
    session degeri = ${sessionScope["adi"]}<br>
    request i yapan kişinin ip si=${pageContext.request.remoteAddr}

</body>
</html>
```

JSP İfade Dili (Expression Language)

İfade dilinde tanımlı değişkenler

pageContext: uygulama genelindeki değişkenler

param: parametreler

header: `ServletRequest.getHeader("ad") = ${header.ad}`

cookie: `${cookie.ad.value}` adı verilen cookinin değerini çevirir

pageScope: sayfadaki değişkenler

requestScope: request nesnesindeki değişkenler

sessionScope: session daki değişkenler

applicationScope: uygulama genelindeki değişkenler

JSTL

(Sun firmasının etiket kütüphanesi)

Sun firmasının geliştirmiş olduğu etiketler temel olarak 4 kütüphanede bulunur.

Tanım	Önek(Prefix)	Varsayılan URI
Core	c	http://java.sun.com/jsp/jstl/core
XML	x	http://java.sun.com/jsp/jstl/xml
Biçimlendirme	fmt	http://java.sun.com/jsp/fmt
Veri Tabanı	sql	http://java.sun.com/jsp/sql

JSTL

(Sun firmasının etiket kütüphanesi)

- ***core** : Temel fonksiyonlar: döngüler, koşul deyimleri, I/O
- ***xml**: XML ile ilgili işlemler
- ***fmt**: para birimi, tarihler gibi biçimlendirme işlemleri
- ***sql**: Veritabanı erişimi

JSTL

(Sun firmasının etiket kütüphanesi)

```
Ornek05.jsp x
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSTL-Java Etiket Kütüphanesi</title>
</head>
<body>
<%
    for (int i=0;i<=10;i++){
    %>
    <%=i%>
    <% }
    %>
    <br>
    <c:forEach var="i" begin="0" end="10">
        <c:out value="{i}" />
    </c:forEach>
</body>
</html>
```

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import="java.util.ArrayList" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSTL - Java Etiket Kütüphaneleri</title>
  </head>
  <body>
    <c:if test="${param.ad != null}">
      <b> Merhaba ${param.ad}! </b><br>
    </c:if>
    <jsp:useBean id="IA" class="java.util.ArrayList" />
    <%
      IA.add("blue");
      IA.add("green");
      IA.add("red");
    %>

    <c:forEach var="aa" items="${IA}">
      <font color="${aa}"><c:out value="${aa}" /></font><br>
    </c:forEach>

  </body>
</html>
```

Üst Düzey Programlama

JSP Etiketletleri(Tags)