

Java Server Pages (JSP)



Java Server Pages (JSP) Nedir?

- JSP öncesinde **HTML sayfa içeriği** Servlet içerisinde oluşturulmaktaydı
- Eğer uygulamanın **HTML layout'unda bir değişiklik** yapmak gerekirse doğrudan Java kodunun değiştirilmesi gerekiyordu
- Uygulamanın iş mantığını oluşturan kod ile web sayfalarının layout'unu oluşturan **kod iç içe** girmekteydi

Problem :Servlet İçinden HTML İçerik Üretilmesi

```
public void service(ServletRequest request, ServletResponse
response){
    PrintWriter writer = ((HttpServletResponse)
response).getWriter();

    writer.write("<html>");
    writer.write("<body>");
    writer.write("<table>");

    for(int i=0; i<10; i++){
        writer.write("<tr><td>");
        writer.write("" + i);
        writer.write("</td></tr>");
    }

    writer.write("</table>");
    writer.write("</body>");
    writer.write("</html>");
}
```

Çözüm :Java Server Pages (JSP)

- JSP ile HTML içeriğin Java kodu içerisinde yazılması **tersine** döndü
- HTML içerik **normal HTML sayfaları** oluşturulur gibi JSP sayfalarında yazılmaya başlandı
- Java kodu ise **kod parçacıkları (scriptlet)** şeklinde **JSP sayfasının içerisinde** yazılabilir hale geldi

Sonuç :JSP İçinde Java Kodu

```
<html>
  <body>
    <table>
      <%
        for(int i=0; i<10; i++){
          <tr>
            <td><%=i%></td>
          </tr>
        <%
      %>
    </table>
  </body>
</html>
```

Ancak HTML içerik ile iş mantığı ile ilgili java kodunun iç içe geçmesi problemi ortadan Kalmamış oldu.

Problem sadece bir noktadan alınıp başka bir noktaya taşındı.

JSP Nasıl Çalışır?

JSP sayfaları Servlet sınıflarına dönüştürülerek Çalıştırılırlar

Bu işlem ilk request'de de gerçekleşebilir

Cevap dönme süresini kısaltmak için Uygulamanın startup aşamasında da yapılabilir



Client Browser

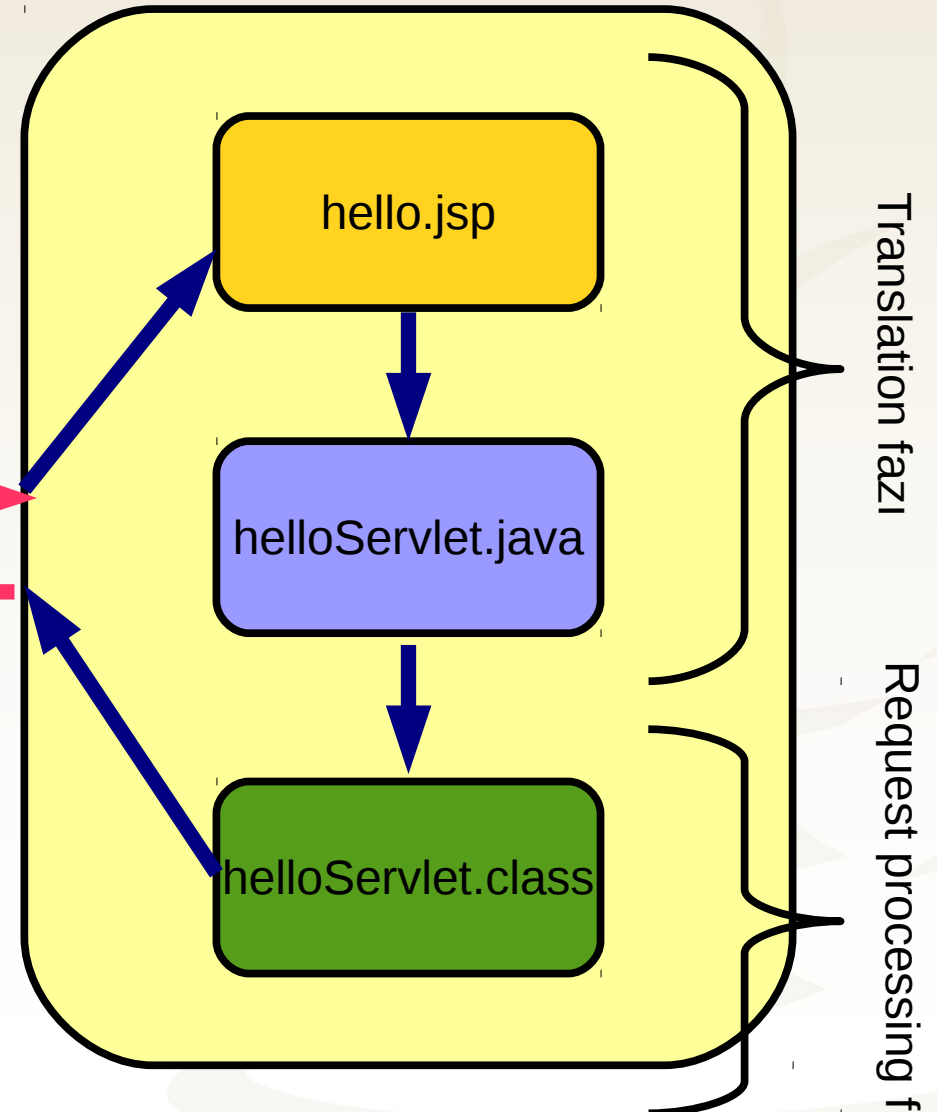
HTTP request: **hello.jsp**

HTTP response:

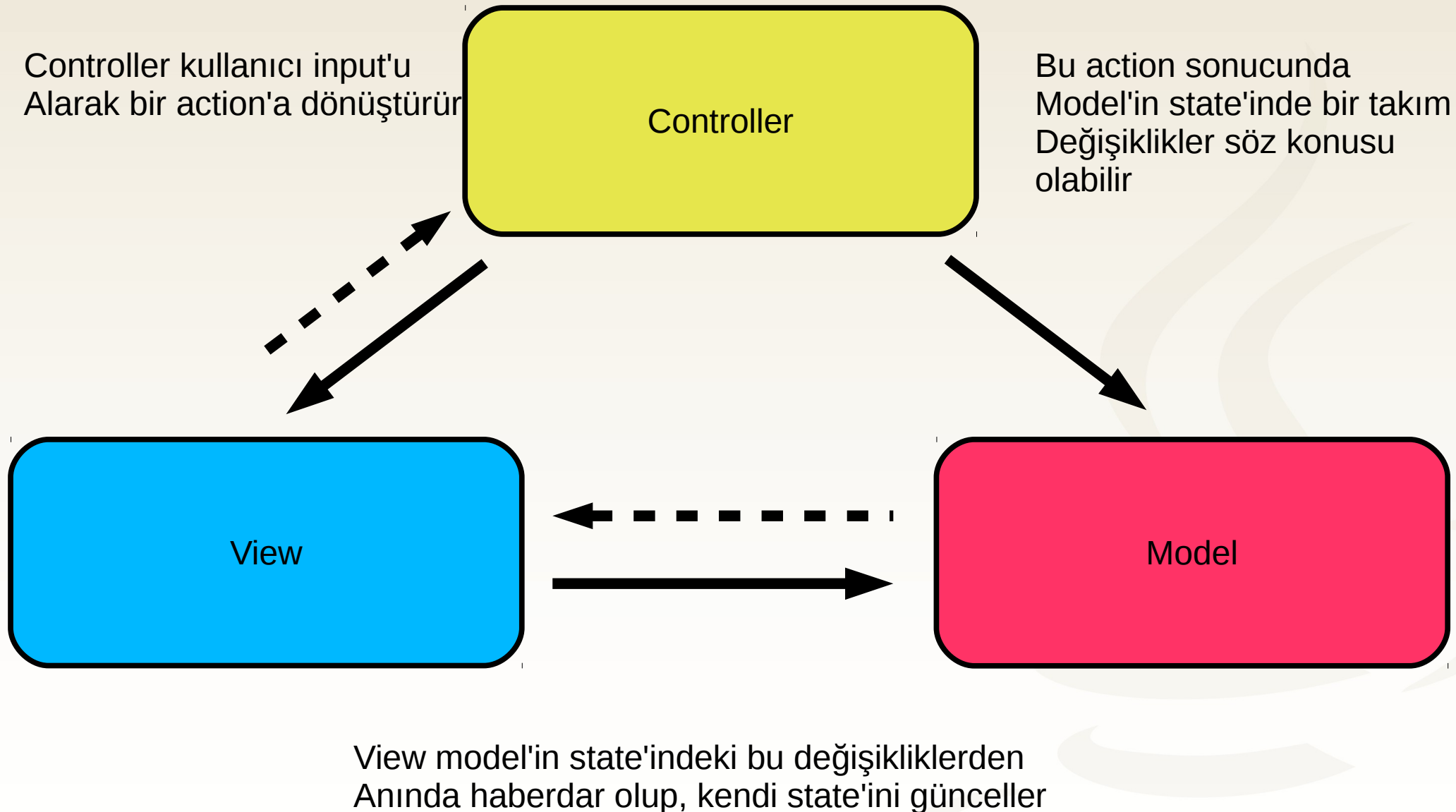
```
<html>
<body>
Hello World!
</body>
</html>
```

JSP sayfasında yapılacak değişiklikler Dinamik olarak yansıtılmaktadır

Web/JSP Container



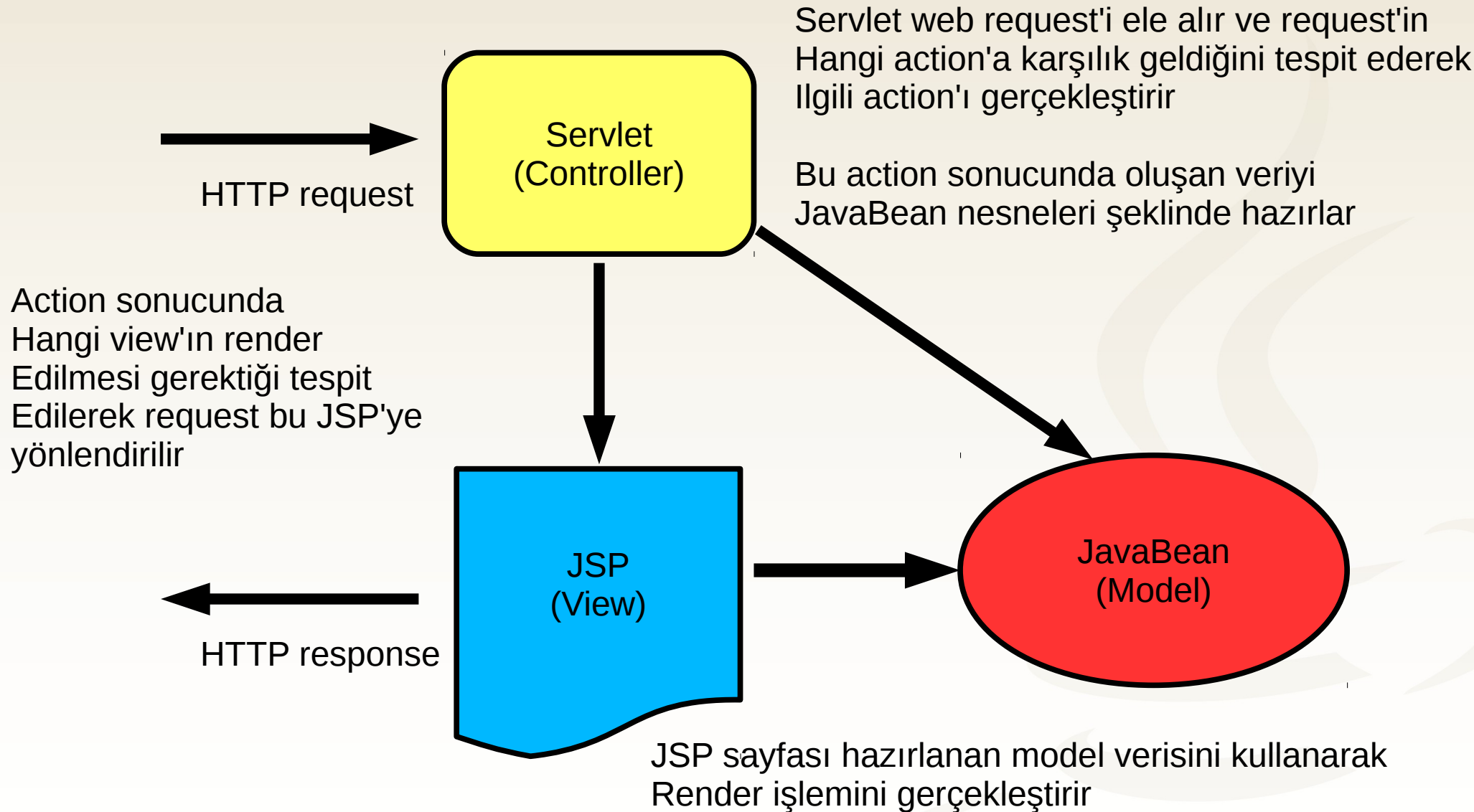
Model View Controller



MVC'nin Web'e Uyarlaması: Model2

- Web uygulamalarının gelişim sürecinde Servlet ve JSP'ler ilk çıktığında **view ve iş mantığının iç içe girdiği çözümler** uygulanmakta idi
- Bu çözümler genellikle **Model 1** olarak adlandırılmıştır
- Tam olarak örtüşmese de MVC'yi temel alan mimarisel bir yaklaşım ile **view ve iş mantığı kısımlarını birbirinden ayıran** Web mimarisine ise **Model 2** adı verilmiştir

Model 2 veya MVC2



JSP Container

- JSP Container çoğunlukla web container içerisinde **bütün jsp request'lerini ele alan bir Servlet instance'**ıdır
- JSP sayfaları derleme aşamasından itibaren **normal bir Servlet** gibi işleme tabi tutulurlar
- Tek fark Servlet'in **dinamik olarak üretilmesidir**
- Request'i asıl ele alan ve response'u üreten aslında JSP'nin servlet karşılığıdır

JSP Örneği

```
<%@page import="java.util.Date"%>
<%!
    String message = "Hello World";
    Date now = new Date();
%>
<html>
    <!-- bu bir commenttir--%>
    <%@ include file="header.jsp" %>
    <body>
        Session :<%=session.getId()%><br/>
        <%
            out.println(message);
            out.println("Current time :" + now);
        %>
        <table>
            <%
                for(int i=0; i<10; i++){
            %>
            <tr>
                <td><%= (3+9)%> - <%=i%></td>
            </tr>
            <%
                }
            %>
        </table>
    </body>
</html>
```

JSP Örneğinin Anatomisi

`<%@page import="java.util.Date"%>` → JSP page directive

`<%! String message = "Hello World"; Date now = new Date();%>`

→ JSP declaration

`<html>`

`<%-- bu bir commenttir--%>` → JSP comment

`<%@ include file="header.jsp" %>` → JSP include directive

`<body>`

Session : `<%=session.getId()%>` `
`

`<%`

`out.println(message);`
`out.println("Current time : " + now);`

JSP scriptlet
Herhangi bir java ifadesi olabilir
Java ifadelerinin sayısında sınır yoktur

`%>`

`<table>`

`<%`

`for(int i=0; i<10; i++){`

`%>`

`<tr>`

`<td><%= (3+9) %> - <%= i %> </td>`

`</tr>`

`<%`

`}`

`%>`

→ JSP implicit objects

JSP expression
Java ifadesi String'e çevrilir

`</table>`

`</body>`

`</html>`

JSP Implicit Objects

- **request** : HttpServletRequest nesnesine erişim sağlar
- **response** : HttpServletResponse nesnesine erişim sağlar
- **out** : PrintWriter nesnesine erişim sağlar
- **session** : HttpSession nesnesine erişim sağlar
- **application** : ServletContext nesnesine erişim sağlar

JSP Implicit Objects

- **config** : ServletConfig nesnesine erişim sağlar
- **page** : Java'daki this anahtar kelimesine karşılık gelir
- **pageContext** : JSP sayfasını temsil eder
 - Diğer implicit nesnelerin bir kısmında bu nesne içerisinde saklanır
 - Attribute'lar PAGE, REQUEST,SESSION ve APPLICATION_SCOPE field'ları ile içerisinde saklanabilir
- **exception** : Bir önceki JSP sayfasında fırlatılan Exception'a erişim sağlar

JSP Directives

- JSP sayfası içerisinde **<%@ directive attribute="value" %>** şeklinde bir ifadedir
- JSP sayfasının **translation fazı** için bir takım **direktifler ve komutlar** verilmesini sağlar
- Üç türüdür
 - page <%@ page attribute="value" %>
 - include <%@ include attribute="value" %>
 - taglib <%@ taglib attribute="value" %>

JSP Scripting Tag'leri

- JSP sayfası içerisine **<% Java code %>** şeklinde yazılan Java kod bloklarıdır
- Bu kod blokları translation aşamasında oluşturulan Java sınıfı içerisinde yer alırlar
- Üç türüdür
 - Scriptlet **<% Java code %>**
 - Decleration **<%! Java code %>**
 - Expression **<%= Java code %>**

JSP Scripting Tag'leri

- **Scriptlet**
 - Doğrudan **_jspService()** metodu içerisine taşınan kod bloğudur
 - Bir veya daha fazla satırdan oluşabilir
- **Declaration**
 - **Sınıf değişkenleri** tanımlamak, **jspInit()**, **jspDestroy()** metotlarını yazmak için kullanılır
- **Expression**
 - Bir java expression'dır, **sonunda ; olmamalıdır**
 - Değeri, **string'e** dönüşür

JSP ve Exception Handling

owners.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" errorPage="error.jsp"%>
<html>
  <body>
    <%
      if(true) throw new RuntimeException("hata meydana geldi");
    %>
  </body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isErrorPage="true"%>
<html>
  <body>
    JSP sayfasında hata meydana geldi:<br/>
    <%
      exception.printStackTrace(response.getWriter());
    %>
  </body>
</html>
```

JSP ve Java Bean Kullanımı

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

<html>
  <body>
    <jsp:useBean id="owner"
class="com.javaegitimleri.petclinic.model.Owner" scope="page" />

    <jsp:setProperty property="firstName" name="owner"
value="Kenan" />

    <jsp:getProperty property="firstName" name="owner" />

  </body>
</html>
```

Scope'da tanımlı bean'ı **bulur** veya **yoksa oluşturur**

Bean property değerini set eder

Bean property değerine erişir

Scope: page, request, session, application olabilir

JSP Expression Language

- JSP ve JSF için ortak ifade dilidir
- JSP sayfalarının içindeki tag'lerin attribute'larında veya doğrudan sayfanın içerisinde **`${expr}`** şeklinde java expression'ları yazılabilir
- Scope değişkenlerine, parametre değerlerine vs erişilebilir
- JSP implicit nesnelerinde de erişilebilir
- Artimatik, mantıksal veya ilişkisel işlemler yapılabilir

JSP Expression Language

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page isELIgnored="false"%>
<html>
  <body>
    <jsp:useBean id="rectangle" class="x.y.z.Rectangle" scope="page" />

    <jsp:setProperty property="width" name="rectangle" value="10"/>

    <jsp:setProperty property="height" name="rectangle" value="20"/>

    <jsp:setProperty name="rectangle" property="area"
value="${rectangle.width*rectangle.height}" />

    Rectangle area:${rectangle.area}<br/>
    Rectangle area:${pageScope.rectangle.area}<br/>

    Current session id:${pageContext.session.id}<br/>
    Context path:${pageContext.request.contextPath}

  </body>
</html>
```

→ `isELIgnored="false"` ifadesi reserved olmadığı için bu karakterlerin kullanıldığı eski jsp sayfalarının yeni jsp container'larda çalışabilmesi için bu directive kullanılabilir

JSP Expression Language

```

<%@ page import="com.javaegitimleri.Kisi" %>
<%@ page isELIgnored="false" %>
<html>
<body>
<jsp:useBean id="kisi" scope="page" class="com.javaegitimleri.Kisi"/>

<jsp:setProperty property="ad" name="kisi" value="kenan"/>

<jsp:setProperty property="soyad" name="kisi" param="soyad"/>

${kisi.soyad}
${kisi[param['propertyName']]}
```

propertyName isimli request param'ın değerine göre
hangi property'ye erişileceğine karar verilebilir

http://localhost/?propertyName=ad&soyad=sevindik

```

<br/>
${kisi.telefonNumaralari[0]}
<br>
${kisi.telefonNumaralari[1]}
```

telefonNumaralari isimli bir List'in elemanlarına erişilebilir

Benzer biçimde map içindeki key:value pairlere de erişilebilir
myMap.key veya myMap['key']

```

</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)