

Java Server Pages (JSP)

Java Server Pages

- **serverseitig dynamisch generierte Webseiten, Schwerpunkt auf HTML-Code (bzw. allgemein XML)**
- **statisches HTML-"Gerüst" mit zusätzlichem JSP-Code, der vor Versand der Seite ausgeführt bzw. ausgewertet wird**
- **diverse vordefinierte Funktionen, Definition eigener Tags, etc.**

hello.jsp

```
<html>
<body>

    <h1>Hello World</h1>

    <% String s = "Max; %>

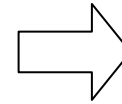
    <p>
        Hier steht noch viel mehr HTML
        Hallo <%= s %> !
    </p>

</body>
</html>
```

Implementierung

- **.jsp-Datei wird von JSP-Compiler in ein Servlet übersetzt (in Java-Quellcode oder direkt zu Bytecode)**
- **"Abstraktion von Servlets"**

```
<html>
<body>
  <% int value = 1; %>
  <table>
    <tr><td><%= value+1 %></td></tr> ...
  </table>
</body></html>
```



```
class _servletclass implements Servlet, HttpJspPage {
  ...
  public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    ServletConfig config = ...;
    Object page = this;
    PageContext pageContext = ...;
    JspWriter out = pageContext.getOut();
    HttpSession session = request.getSession( true );
    try {
      out.print( "<html>\r\n" );  out.print( "<head>\r\n" );
      int value = 1;
      out.print( "<table>\r\n" );  out.print( " <tr><td>" );
      out.print( toStringOrBlank( val + 1 ) );
      ...
    } catch ( Exception _exception ) {
      [Weiterleiten an bestimmte Fehlerseite]
    }
  }
}
```

Einbetten von Java-Code in statisches HTML

- **Attribut der generierten Servlet-Klasse**

```
<%! int attribute1 = 2; %>
```

- **Rumpf der service-Methode**

```
- <% int tmp = 3; %>
```

```
- <% if (rabatt) { %>
```

```
    <b><i>jetzt reduziert!</i></b>
```

```
<% } %>
```

- **zu Zeichenkette ausgewerteter Ausdruck**

```
- <p>die Summe beträgt
```

```
<%= sum*(1-rabatt) %>
```

```
Euro</p>
```

Vordefinierte Variablen

- `PageContext pageContext`
- `HttpRequest request`
- `HttpResponse response`
- `HttpSession session`
- `JSPWriter out`
- `Servlet page`

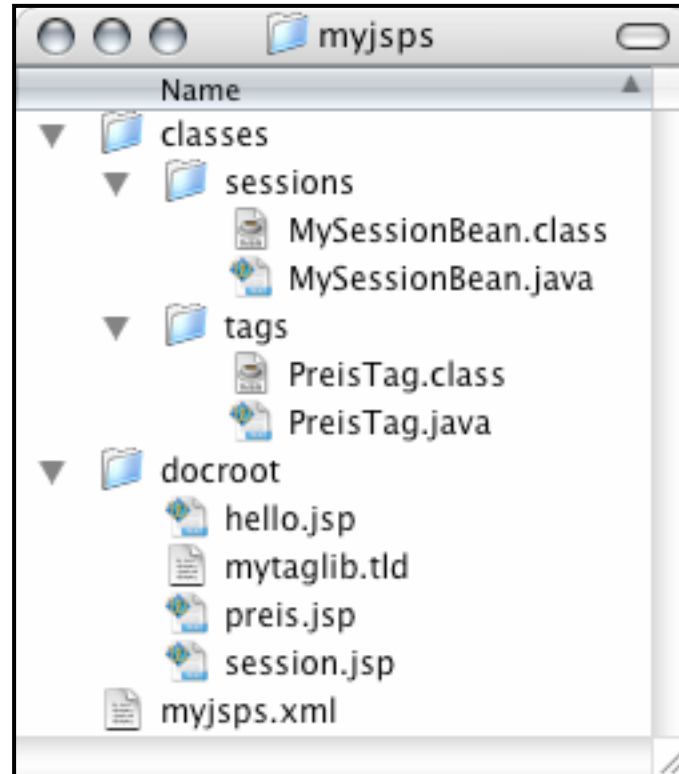
JSP-Direktiven

- **include: ähnlich z.B. C-Präprozessor**
 - `<%@ include file="impresum.html" %>`
- **page: Einstellungen für die Seite/Servlet-Klasse**
 - **Import von Java-Klassen**
`<%@ page import="project.general.*" %>`
 - **MIME-Type festlegen**
`<%@ page contentType="text/html" %>`
- **taglib: Einbinden (eigener) Tag-Bibliotheken**
 - `<%@ taglib prefix="my" uri="lib/mytags.tld" %>`

Kombination mehrerer JSP-Seiten

- `jsp:forward`
 - **JSP-Seite erzeugt Teil der HTML-Seite und leitet Anfrage dann an andere JSP-Seite weiter**
 - ```
<jsp:forward page="fussteile.jsp" >
 <jsp:param name="summe" value="73.95" />
</jsp:forward>
```
- `jsp:include`
  - **Seite übergibt Anfrage "kurzzeitig" an andere, bekommt dann aber Kontrolle/Anfrage zurück (ähnlich include-Direktive, hier jedoch gültige jsp-Seite, kein Code-Fragment)**
  - ```
<jsp:include page="warenkorb.jsp" >  
    <jsp:param name="rabatt" value="0.05" />  
</jsp:include>
```

Beispiele (Jetty)



myjsps.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN" "http://jetty.mortbay.org/configure.dtd">
<Configure class="org.mortbay.jetty.Server">

  <Call name="addListener">
    <Arg>
      <New class="org.mortbay.http.SocketListener">
        <Set name="Port"><SystemProperty name="jetty.port" default="8080"/></Set>
      </New>
    </Arg>
  </Call>

  <Call name="addContext">
    <Arg>/jsp/*</Arg>

    <Call name="addServlet">
      <Arg>JSP</Arg>
      <Arg>*.jsp</Arg>
      <Arg>org.apache.jasper.servlet.JspServlet</Arg>
    </Call>

    <Set name="ResourceBase">/home/datsche/haustein/myjsps/docroot</Set>
    <Set name="ClassPath">/home/datsche/haustein/myjsps/classes</Set>
  </Call>

</Configure>
```

Benutzer-Identifikation/Speicherung eines Zustands: MySessionBean.java

```
package sessions;

import java.util.*;

public class MySessionBean {

    int counter;
    boolean reset;

    public int getCounter() {
        if(reset) counter = 0;
        return counter++;
    }

    public void setReset(String reset) {
        this.reset = "true".equals(reset);
    }
}
```

session.jsp

```
<%@ page import = "sessions.MySessionBean" %>

<jsp:useBean id="state" class="sessions.MySessionBean" scope="session"/>
<jsp:setProperty name="state" property="*" />
    <!-- http-request parameter name match -->

<html>
<head><title>Counter</title></head>
<body>

    Counter: <%= state.getCounter() %>
    <br>
    <a href="session.jsp?reset=false">inc</a>
    <a href="session.jsp?reset=true">reset</a>

</body>
</html>
```

preis.jsp

```
<html>
<body>
<%@ taglib prefix="mytags" uri="/mytaglib.tld" %>

<h2>Top-Angebote:</h2>

<p>DVD-Player<br/>
<mytags:preis preis="100" rabatt="0.05"></mytags:preis>
</p>

<p>Video-Recorder<br/>
<mytags:preis preis="50" rabatt="0.03"></mytags:preis>
</p>

<p>Schlagen Sie zu!</p>

</body>
</html>
```

PreisTag.java

```
import javax.servlet.jsp.*; import javax.servlet.jsp.tagext.*;
import java.io.IOException;

public class PreisTag extends SimpleTagSupport {
    int preis = 0;
    float rabatt = 0;

    public void doTag() throws JspException, IOException{
        StringBuffer sb = new StringBuffer("Preis: ");
        sb.append(preis);
        if (rabatt > 0) {
            sb.append(" jetzt für nur <b>");
            sb.append(Math.round(preis * (1 - rabatt)));
            sb.append("</b>");
        }
        getJspContext().getOut().write(sb.toString());
    }

    public void setPreis(String preis) {
        this.preis = Integer.parseInt(preis);
    }

    public void setRabatt(String rabatt) {
        this.rabatt = Float.parseFloat(rabatt);
    }
}
```

mytaglib.tld

```
<?xml version="1.0" encoding="UTF-8" ?>
<taglib xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_0.xsd"
  version="2.0">
  <description>bla</description>
  <tlib-version>1.0</tlib-version>
  <short-name>bla</short-name>
  <uri>/MyTagLib</uri>
  <tag>
    <description>bla</description>
    <name>preis</name>
    <tag-class>tags.PreisTag</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>preis</name>
      <required>true</required>
      <fragment>false</fragment>
    </attribute>
    <attribute>
      <name>rabatt</name>
      <required>true</required>
      <fragment>false</fragment>
    </attribute>
  </tag>
</taglib>
```