



Web- og serverprogrammering



Java Server Pages - dag 2

Java Server Pages fortsat
Implicit definerede objekter. Sessioner
Evt.: JSTL (Java Standard Template Library)

Læsning: WJSP kapitel 3 - 3.5, 4 - 4.2



Formularer og interaktive sider



1) Klienten laver en forespørgsel

2) Serveren sender en HTML-side med formular

F.eks `<form action="side2.jsp"><input type="tekst" name="alder"> <input type="submit" name="send" value="OK"></form>`

1) Klienten viser HTML-side med formular

2) Bruger udfylder formular og trykker 'OK'

3) Klienten laver en ny forespørgsel med formular

F.eks. `GET /side2.jsp?alder=34&send=OK`

1) Serveren fortolker formulardata

`request.getParameter("alder")` er "34" (og `req.getP("send")` er "OK")

1) Serveren sender en ny HTML-side

Indholdet afhænger af (sædvanligvis) af de indsendte formulardata



Formularer og interaktive sider



```


  
og din kode (kodefelt):



```

Typer af formularfelter

Skriv dit navn (tekstfelt):

og din kode (kodefelt):

Beskriv dig selv (tekstområde):

Hvad foretrækker du at programmere i (radioknapper):

C C++ Java

Hvad kan du programmere i (afkrydsningsfelter):

C C++ Java

Hvilken ret foretrækker du (valgliste):

Hvilke retter kan du spise (valgliste):

- Spaghetti med kødsovs
- Pizza
- Ostefondue
- Rugbrød

```

<p>Beskriv dig selv (tekstområde): <br>
<textarea name="beskrivelse" rows="2" cols="30">Jeg taler esperanto
</textarea>
</p>

<p>Hvad foretrækker du at programmere i<br>(radioknapper):
<input type="radio" name="foretr_prg" value="c">C
<input type="radio" name="foretr_prg" value="cpp">C++
<input type="radio" name="foretr_prg" value="java" checked="checked">Java
</p>

<p>Hvad kan du programmere i<br>(afkrydsningsfelter):
<input type="checkbox" name="kan_prg" value="cpp">C++
<input type="checkbox" name="kan_prg" value="c" checked="checked">C
</p>
<input type="checkbox" name="kan_prg" value="java" checked="checked">Java

<select name="foretr_spise">
<p>Hvilken ret foretrækker du (valgliste): <br>
<option>Spaghetti med kødsovs</option>
<option selected="selected">Pizza</option>
<option>Ostefondue</option>
<option>Rugbrød</option>
</select>
</p>

<p>Hvilke retter kan du spise (valgliste): <br>

```



Formularer og interaktive sider



The image shows a web browser window titled "Parametre3 - Konqueror" displaying the output of a Java servlet. The browser's address bar shows the URL: `http://localhost:8080/kan_spise=piz&kan_spise=rug&indsendKnap=Indsend`. The browser's status bar at the bottom indicates "Side indlæst." (Page loaded).

The browser content displays the following text:

```
Her er parameteren "navn": Jacob

Andre parametre:
kode med værdi: 'abcdef'
id med værdi: '1234'
foretr_spise med værdi: 'Pizza'
foretr_prg med værdi: 'java'
navn med værdi: 'Jacob'
kan_prg med værdi: 'c'
indsendKnap med værdi: 'Indsend data'
kan_spise med værdi: 'spa'
beskrivelse med værdi: 'Jeg taler esperanto'

Parameteren 'kan_spise' har flere værdier.
De er:
De er: spa piz rug
```

The background of the image shows the source code of the servlet, which is a Java class named `Parametre3` extending `HttpServlet`. The code uses `HttpServletRequest` to retrieve parameters and `HttpServletResponse` to write the output. The HTML output is generated using `out.println()` and `out.print()` methods. The code includes comments in Danish and uses `Enumeration` to iterate over all parameters and `getParameterValues()` to handle multiple values for the `kan_spise` parameter.

```
<html>
<head><title>Parametre3</title></head>
<body>
<p>
Her er parameteren "navn": <%= request.getParameter("navn") %><br>
</p>
<p>
Andre parametre:<br>
<%
// udskriv alle parametrene
java.util.Enumeration enum = request.getParameterNames();
while (enum.hasMoreElements()) {
String parameternavn = (String)enum.nextElement();
String parameterværdi = request.getParameter(parameternavn);
out.println(parameternavn+" med værdi: '"+parameterværdi+"'<br>");
}
%>
</p>
<p>
Parameteren 'kan_spise' har flere værdier.<br>
De er:
<%
String[] værdier = request.getParameterValues("kan_spise");
if (værdier != null) {
for (int i = 0; i < værdier.length; i++)
out.println(værdier[i]);
}
</p>
%>
```



Test hinanden



Udføres med naboen (vejledende svar findes i afsnit 2.7 og 3.5)

- Hvad er HTML-kode? Hvor udføres det (klienten eller serveren)?
- Hvad er JSP-programkode? Hvor udføres det (klienten eller serveren)?
- Hvad kan man skrive mellem `<% og %>`?
- Hvad kan man skrive mellem `<%= og %>`?
- Hvad kan man skrive mellem `<%-- og --%>`?
- Hvad kan man med `out`-objektet?
- Hvad kan man med `request`-objektet?
- Hvad er en parameter? Giv et eksempel. Hvordan aflæser man den?
- Hvad er en HTML-formular?
- Giv et lille eksempel på HTML-koden i en formular.
- Beskriv hvordan samspillet mellem klient og server foregår - beskriv de parametre der kommer fra dit eksempel på en formular.
- Nævn så mange typer felter, der kan forekomme i en formular, som du kan komme i tanke om.



Projektopgaven



- Erfaring med webprogrammering gennem et projekt
 - Omfang: Cirka 80 timer
 - Formål:
 - at deltagerne får programmeret så meget som muligt på et projekt
 - at de får arbejdet med de ting fra kurset der interesserer dem mest
 - en mindre del af de berørte teknikker i kurset skal indgå
 - Kan laves i grupper á 1-3 personer
- Det kunne f.eks. være:
 - et projekt fra arbejdet
 - en prototype på dit afgangsprøveprojekt
 - en interaktiv webside du savner
 - Diskussionsforum, foreningsside, gmail-klon, e-handel, ...
 - Et elektronisk "dueslag" til aflevering af skriftlige opgaver i folkeskolen!
- Du skal lægge dig fast på et projekt til gang 4
 - Hjemmeopgave: Beskriv din ide til projekt (10-30 linier).



Projektopgaven



Opgave til denne gang

1. Bestem hvilken webapplikation du godt kunne tænke dig at programmere
2. Lav en oversigt over skærbillederne der skal være i din webapplikation
3. Notér de variationer der kan være i hvert skærbillede i forskellige situationer (f.eks. ved forkert indtastede data)
4. Lav så mange af HTML-siderne du kan (brug f.eks. Netscape's HTML-Composer, JDeveloper eller Macromedia Dreamweaver, hvis du er så heldig at have den).



Mere om JSP: Sessioner



- Hver bruger får tildelt et session-objekt når de besøger en JSP-side.
- Sessionen følger brugeren, lige meget hvilken side han/hun er inde på, og er derfor nyttigt til at huske data, der skal følge brugeren.



Mere om JSP: Sessioner

```
<h3>Skriv et ønske</h3>
```

```
Skriv noget, du ønsker.
```

```
<form>
```

```
<input type="text" name="oenske">
```

```
</form>
```

```
<%
```

```
    // hent listen over ønsker
```

```
    ArrayList liste = (ArrayList) session.getAttribute("ønsker");
```

```
    if (liste == null) {                                // hvis listen ikke findes:
```

```
        liste = new ArrayList();                       // opret den
```

```
        session.setAttribute("ønsker", liste); // og registrer den under "ønsker"
```

```
    }
```

```
    // se om der kommer en parameter med endnu et ønske
```

```
    String ønske = request.getParameter("oenske");
```

```
    if (ønske != null) {
```

```
        liste.add(ønske);                               // tilføj ønske
```

```
    }
```

```
    if (liste.size()>0) {
```

```
        %>
```

```
        <h3>Ønskeseddel</h3>
```

```
        Indtil nu har du følgende ønsker:<br />
```

```
        <%
```

```
        // udskriv hele listen
```

```
        for (int i=0; i<liste.size(); i++)
```

```
        { %>
```

```
            Ønske nr. <%= i %>: <%= liste.get(i) %><br />
```

```
        }
```

```
        <% }
```

```
%>
```

oenskeseddel - Konqueror

Eil Redigér Vjs Kør Bogmærker Værktøjer Opsætning Vindue Hjælp

Sted: /kapitel_05/oenskeseddel.jsp?oenske=Et+stort+hus

Dette eksempel demonstrerer, hvordan session-objektet kan bringes til at huske brugerens indtastninger.

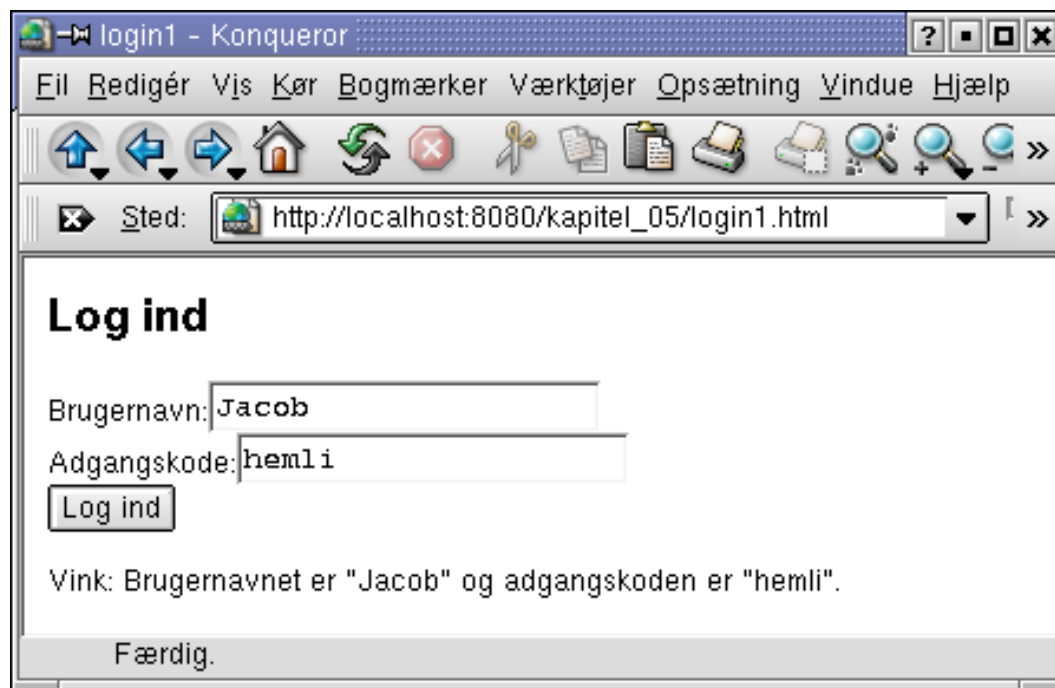
Skriv et ønske

Skriv noget, du ønsker.

```
<html>
<head><title>login1</title></head>
<body>
<h1>Log ind</h1>

<form method="post" action="login2.jsp">
Brugernavn:<input type="text" name="brugernavn" /><br />
Adgangskode:<input type="password" name="adgangskode" /><br />
<input type="submit" value="Log ind" />
</form>
<p>
Vink: Brugernavnet er "Jacob" og adgangskoden er "hemli".
</p>
</body>
</html>
```

Eksempel: Login

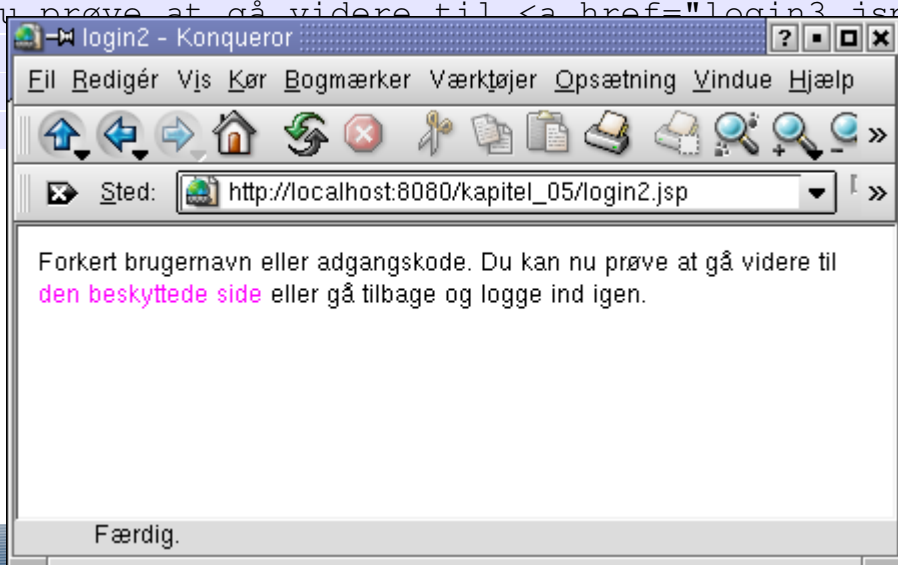




```
<html>
<body>
<head><title>login2</title></head>
<%

// dette burde selvfølgelig hentes fra en database eller lign.
// hvis brugernavn="Jacob" og adgangskode="hemli" logges der ind.
    "hemli".equals(request.getParameter("adgangskode"))
if ("Jacob".equals(request.getParameter("brugernavn")) &&
{
    // sæt attributten "logget ind" i sessionen
    out.println("Du er logget korrekt ind.");
    session.setAttribute("logget ind", "ja");
}
else
    session.removeAttribute("logget ind");
    // fjern attributten "logget ind" fra sessionen
}
    out.println("Forkert brugernavn eller adgangskode.");

%>
eller gå tilbage og logge ind igen.
Du kan nu prøve at gå videre til <a href="login3.jsp"
</body>
</html>
```





```
<%  
// se om attributten "logget ind" er sat i sessionen  
if (session.getAttribute("logget ind") == null) {  
    // brugeren er ikke logget ind, så send ham tilbage til login-siden  
    response.sendRedirect("login1.html");  
}  
%>  
<html>  
<head><title>login3</title></head>  
<body>  
  
<h1>Den beskyttede side</h1>  
Denne tekst kan du kun se, hvis du er logget korrekt på.  
  
</body>  
</html>
```





Implicit definerede objekter



- Der findes en række implicit definerede objekter, som man altid har adgang til i en JSP-side:
 - **request** - anmodningen fra klienten
 - **response** - svaret til klienten
 - **out** - skrive tekst til klienten
 - **session** - objekt der følger den enkelte bruger
 - **application** - fælles for hele webapplikationen
 - **logging**
 - **konfigurations-parametre fra web.xml**
 - **kan også gemme attributter ligesom session-objektet**
 - **config** - den enkelte websides konfiguration
 - **page** - selve JSP-siden
 - **exception** - undtagelse opstået under kørsel
 - **pageContext** - alle objekterne samlet i ét



Java Standard Template Library



- JSTL (JSP Standard Tag Library)
 - Et tag library (forkortet taglib) er et bibliotek af HTML-lignende koder.
 - Ligesom JSP-koderne udføres taglib-koderne på serveren.
- Udkom i slutningen af 2003,
- Muligt at programmere JSP-sider i rent JSTL

```
<!-- <%@ taglib uri="http://java.sun.com/jstl/core" prefix="c"%> -->
<html>
<head><title>Alder med JSTL</title></head>
<body>
<p>
<c:set var="alder" value="31" />
Søren er <c:out value="${alder}" /> år gammel.
Det svarer til <c:out value="${12*alder}" /> måneder.<br />

<c:set var="alder" value="3" />
Julie er <c:out value="${alder}" /> år gammel.

<c:set var="alder" value="${alder*365}" />
Det svarer cirka til <c:out value="${12*alder}" /> dage.<br />
</p>
</body>
</html>
```



Java Standard Template Library



```
<!-- <%@ taglib uri="http://java.sun.com/jstl/core" prefix="c"%> -->
<html>
<head><title>Syvtabellen med JSTL</title></head>
<body>
<p>Her er syv-tabellen:<br />
<b><c:forEach var="i" begin="1" end="10">
  Syv gange <c:out value="{i}" /> er: <c:out value="{7*i}" />.<br />
</c:forEach>
</p>
</body>
</html>
```

```
<!-- <%@ taglib uri="http://java.sun.com/jstl/core" prefix="c"%> -->
<html>
<head><title>Løkker med JSTL</title></head>
<body>
<p>Her er nogle frugter:<br />
  <c:forEach var="f" items="æble,pære,banan,blomme">
    En frugt: <c:out value="{f}" /><br />
  </c:forEach>
</p>
</body>
</html>
```



JSTL og formularer



Parametre3 - Konqueror

Fil Redigér Vjs Kør Bogmærker Værktøjer Opsætning

Indsend data & kan_spise=piz&kan_spise=rug&indsendKnap

Her er parametren "navn": Jacob
Parametren på en anden måde: Jacob

Andre parametre:
indsendKnap med værdi: 'Indsend data'
navn med værdi: 'Jacob'
kan_prg med værdi: 'c'
kode med værdi: ''
foretr_spise med værdi: 'Pizza'
kan_spise med værdi: 'spa'
beskrivelse med værdi: 'Jeg taler esperanto '
id med værdi: '1234'
foretr_prg med værdi: 'java'

Parametren 'kan_spise' kan have flere værdier.
De er: spa piz rug

Side indlæst.

```
<!-- <%@ taglib uri="http://java.sun.com/jstl/core" prefix="c"%> -->
<html>
<body>
<head><title>Parametre3 med JSTL</title></head>
<p>
Parametren på en anden måde: <c:out value="{param.navn}" /><br />
Her er parametren "navn": <c:out value="{param['navn']}" /><br />
</p>
Andre parametre:<br />
<p>
<c:out value="{p.key} med værdi: '{p.value}'" /><br />
<c:forEach var="p" items="{param}">
</p>
</c:forEach>
<p>
De er:
Parametren 'kan_spise' har flere værdier.<br />
<c:out value="{pv}" />
<c:forEach items="{paramValues['kan_spise']}" var="pv">
</p>
</c:forEach>
</body>
</html>
```




Åben Dokumentlicens



- Dette foredragsmateriale er under Åben Dokumentlicens (ÅDL)
 - Du har derfor lov til frit at kopiere dette værk
 - Bruger du dele af værket i et nyt værk, skal de dele, der stammer fra dette værk, igen frigives under ÅDL
 - Den fulde licens kan ses på <http://www.sslua.dk/linuxboa/licens.html>

