



# Web- og serverprogrammering



Konfiguration - dag 11

Fremlæggelse af og status for projekter

Konfiguration af Tomcat

web.xml

Pakning og idriftsættelse af webapplikation

Evt.: Integration af Tomcat med Apache

Evt.: Webapplikationer der kører på klienten - gmail

Lærer: JANO

Læsning:WJSP 7



# Eksamensprojektet



## Fremlæggelse af og status for projekter

- Henrik: Fotoklub
- Kim D: Netbanken
- Anne
- Lis: Fraværsregistrering
- Bjarne H + Christian: Dueslag
- Bjarne J: Produktionsstoptidsregistrering (periode, årsag, rapporter)
- Kim J
- Ole Hedevang: Webbutik
- Tommy: Kajakklub
- Jesper: Online pizzeria
- Sven-Erik og Morten: E-handel (nok tøjbutik)
- Preben: Konfigurator (en slags e-handel)
- Svend
- Klaus Elmquist: Procedurale mønstre
- Per: Gruppekalendar
- Stefan: Bowlingklub
- Ratha: Internet-butik om blomster



# Installation af webapplikation



- WAR-fil (Web ARchive)
  - = ZIP-fil med hel applikation
- Hent WAR-fil
- Læg i webapps/ og den installeres automatisk
- Al opsætning findes i WEB-INF/web.xml
  
- Hente parametre fra web.xml
  - application - fælles for alle sider
  - config - for en enkelt side



# Installation af webapplikation



- WAR-fil (Web ARchive)
  - = ZIP-fil med hel applikation
- Al opsætning findes i WEB-INF/web.xml
  
- Installation
  - Lokalt: Læg i webapps/ og den installeres automatisk
  - Pingo: Åbn <http://pingo.cv.ihk.dk:8080/manager/html/> og angiv brugerID stud og adgangskode studx1.
    - Gå ned til "Upload a WAR file to install" og installér
    - MySQL findes på pingo – se i øvelserne hvordan data overføres til Pingo



# Hente parametre fra web.xml



application-objektet: konfiguration fælles for alle sider.

Lave en indgang af typen context-param i web.xml:

```
<web-app>
  ...
  <context-param>
    <param-name>navnet</param-name>
    <param-value>værdien</param-value>
  </context-param>
  ...
</web-app>
```

Nu vil man i en JSP-side kunne hente værdien af initialiseringsparameteren med:

```
String værdi = application.getInitParameter("navnet")
```

hvorefter strengen vil have indholdet 'værdien'.

Fra en servlet er application-objektet er tilgængeligt med:

```
ServletContext application = getServletContext();
```

hvorefter værdien fås som i en JSP-side med:

```
String værdi = application.getInitParameter("navnet")
```



# Hente parametre fra web.xml



config-objektet: konfiguration for den enkelte JSP-side eller servlet.  
Lave en indgang af typen context-param i web.xml under en side:

```
<web-app>
  <servlet>
    <servlet-name>En simpel servlet</servlet-name>
    <servlet-class>SimpelServlet</servlet-class>
    <init-param>
      <param-name>navnet</param-name>
      <param-value>værdien</param-value>
    </init-param>
  </servlet>

  <servlet>
    <servlet-name>jspside</servlet-name>
    <jsp-file>/jspside.jsp</jsp-file>
    <init-param>
      <param-name>navnet</param-name>
      <param-value>værdien</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>En simpel servlet</servlet-name>
    <url-pattern>/servlet/SimpelServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

Fra servleten SimpleServlet kunne dette hentes med

```
String værdi = getInitParameter("navnet");
```

Fra JSP-siden jspside.jsp ville det kunne dette hentes med

```
config.getInitParameter("navnet")
```



# web.xml - URL'er til websider



config-objektet: konfiguration for den enkelte JSP-side eller servlet.

Lave en indgang af typen context-param i web.xml under en side:

```
<web-app>
  <servlet>
    <servlet-name>En simpel servlet</servlet-name>
    <servlet-class>SimpelServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>En simpel servlet</servlet-name>
    <url-pattern>/servlet/SimpelServlet</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>En simpel servlet</servlet-name>
    <url-pattern>/simpel/*</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>jspside</servlet-name>
    <jsp-file>/jspside.jsp</jsp-file>
  </servlet>

  <servlet-mapping>
    <servlet-name>jspside</servlet-name>
    <url-pattern>/jspsider/*</url-pattern>
  </servlet-mapping>
</web-app>
```

# Pakning af webapplikation

- Filstruktur:

0	09-27-03	12:23	WEB-INF/
0	09-27-03	12:23	WEB-INF/lib/
158892	08-13-01	21:42	WEB-INF/lib/log4j.jar
56691	07-16-03	18:34	WEB-INF/lib/oscache.jar
218325	09-27-03	12:23	WEB-INF/lib/JSPWiki.jar
14762	07-16-03	18:44	WEB-INF/jspwiki.properties
9524	08-13-03	23:04	WEB-INF/jspwiki.tld
4031	11-06-02	21:22	WEB-INF/web.xml
2697	08-13-03	22:58	Diff.jsp
6169	03-08-03	11:49	Edit.jsp
1784	03-13-03	23:29	Error.jsp
1572	02-17-03	12:49	PageInfo.jsp
2212	02-17-03	12:49	PageModified.jsp
1529	03-29-03	11:15	Preview.jsp
1802	02-17-03	12:49	Search.jsp
1508	02-17-03	12:49	Upload.jsp
2159	02-17-03	12:49	UserPreferences.jsp
2095	05-23-03	00:37	Wiki.jsp
0	09-27-03	12:02	images/
842	02-17-03	12:49	images/attachment_big.png
178	02-17-03	12:49	images/attachment_small.png
927	05-23-03	00:01	images/out.png
396	06-09-02	15:33	images/xml.png
1182	06-09-02	15:33	images/xmlCoffeeCup.png





# Opsætning af Apache



- Linux
  - redigere /etc/httpd/conf/\*



# Samspil mellem Apache og Tomcat



- Godt ved stor serverbelastning/høj sikkerhed
  - Apache klarer statisk indhold (f.eks. billeder)
  - Apache bruger Tomcat til dynamisk indhold
- Lidt besværligt at sætte op
  - Gør det kun hvis det er nødvendigt
  - Flere forskellige måder det kan ske på (connectors)
- Apache-konfiguration (eksempel)
  - Find mod\_jk2 et sted, f.eks. [www.jpackage.org](http://www.jpackage.org)
  - Indlæs modul fra Apaches httpd2.conf
  - Redigér workers2.properties
    - Angiv hvilke stier Tomcat skal tage sig af



# 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.sslug.dk/linuxbog/licens.html>

