



Denne guide er oprindeligt udgivet på Eksperten.dk

Web applikationer i Java

Denne artikel introducerer web applikationer i Java.

Den beskriver teknologien, forklarer hvilket software man kan bruge, viser hvordan man kan arbejde og giver links til mere information.

Det er ikke en egentlig tutorial.

Skrevet den **28. Feb 2009** af **arne_v** | kategorien **Programmering / JSP** | ★★★★★

Historie:

V1.0 - 12/01/2004 - original

V1.1 - 31/01/2004 - forbedret formatering + lidt mere om hvad det er godt til

Introduktion

Java teknologi dækker efterhånden de fleste software områder. Og selvfølgelig har Java også et bud på web applikationer.

De to Java standarder på dette område er:

- * servlets, som er rene Java kode moduler i stil med ISAPI og NSAPI
- * JSP (Java Server Pages) som er HTML sider med embedded kode ligesom ASP og PHP (det er bare Java kode i <% %>)

JSP er selvfølgelig mere brugt end servlets idag, men der er to gode grunde til at lære servlets alligevel:

- * der er enkelte situationer hvor en servlet er bedre end en JSP side
- * de fleste fif man lærer i forbindelse med servlets kan umiddelbart også bruges med JSP

JSP og servlet udgør sammen med EJB (Enterprise Java Beans) og JCA (Java Connector Architecture) det man kalder J2EE (Java 2 Enterprise Edition).

JSP og servlet er nok lidt sværere at komme igang med end ASP og PHP. De forudsætter at man kan Java. Der er forholdsvis mange ting som man skal sætte sig ind i for at kunne komme igang.

Til gengæld er det en teknologi som er logisk opbygget og designet til at skalere til store løsninger.

JSP/Servlet bruges som oftest til integrerede e-business løsninger sammen med resten af J2EE (EJB og JCA).

Der er forholdsvis få af de web hoteller der henvender sig til private og mindre firmaer som understøtter JSP og servlets.

Dem der bruger J2EE teknologi hoster som oftest selv.

Simple eksempler

Vi tager lige to små eksempler for at illustrere teknologien.

HelloWorldServlet.java

```
package test;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.net.*;
import java.util.*;

public class HelloWorldServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.println("Hello world - time is " + (new Date()) + " !");
    }
}
```

HelloWorld.jsp

```
<%@page import="java.util.*"%>
Hello world - time is <%= (new Date()) %> !
```

Hvordan kommer man igang

Der er adskillige muligheder for server. Komplette J2EE applikations servere: IBM WebSphere, BEA WebLogic, JBoss, Oracle, SUN, Pramati, Borland Enterprise Server, Macromedia JRun. JSP/Servlet engine: Tomcat, Caucho Resin, neddrolede versioner af IBM/BEA/Borland J2EE applikations servere.

Tomcat er den mest brugte, er og er gratis til både udvikling og produktion. Så du går ikke helt galt med den.

En meget vigtig ting ved Java web applikationer er directory strukturen. Den skal være på en bestemt måde ellers virker applikationen ikke.

En typisk directory struktur ser ud som:

```
minapplikation
  *.jsp
  WEB-INF
    web.xml
    classes
    minpakke
```

```
*Servlet.class
*Bean.class
lib
*.jar
```

Man kan normalt enten deploye ved at kopiere minapplikation directory ind under serverens webapps directory eller ved at kopiere en war fil til serverens webapps directory. En war fil er bare en jar fil med ovenstående directory struktur.

web.xml indeholder web applikationens konfiguration. Man kan putte meget i den, men man kan starte med en fil som:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <servlet>
    <servlet-name>HelloWorldServlet</servlet-name>
    <servlet-class>test.HelloWorldServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorldServlet</servlet-name>
    <url-pattern>/HelloWorldServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

En bean er en almindelig Java klasse som kan bruges fra servlet og JSP. En utility klasse. For at være en ægte bean klasse skal den opfylde nogle formelle betingelser, men det er sjældent noget problem.

En servlet skal man selv compile inden man deployer, mens JSP compiles af serveren.

Det er værd at notere sig at JSP dobbelt compiles. Først compiler jspc fra JSP til servlet og så compiler javac fra servlet til binær byte kode. Det har to afledte effekter. For det første kræver JSP at man har en fuld Java SDK, fordi man skal bruge javac. For det andet kan man genbruge en masse servlet kode teknik i JSP, fordi JSP netop compiles til en servlet.

Det kan anbefales at bygge sine war filer med det tool som hedder ant. Det kender strukturen af det hele og sikrer at det bliver pakket rigtigt.

Vigtige tips:

- undlad at installere Java SDK eller Tomcat i directory med mellemrum i da det kan give mange problemer

- put altid både servlets og beans i en pakke da brug af default pakke kan give mange problemer

For at komme igang med JSP og servlets kan man læse bl.a.:

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/> (chapter 11-16)
<http://csajsp-chapters.corewebprogramming.com/Core-Servlets-and-JSP.pdf>

Men hvis du skal seriøst igang med JSP og servlet skal du nok ud og købe nogle bøger.

En fornuftig række følge kunne være:

- simple servlets
- simple JSP
- brug af beans fra servlet og JSP
- brug af container managed form based security
- bruge af container managed database connection pool
- standard tag library
- egne tag libraries
- expression language
- web frameworks som Struts og Webwork

Jeg vil muligvis senere skrive en eller flere artikler om disse emner.

Kommentar af simonvalter d. 14. Jan 2004 | 1

fin intro, flere artikler om emnet er velkomne :)

Kommentar af repsak d. 12. Jan 2004 | 2

Giver et udemærket overblik over servlet/JSP. Savner dog nogle flere argumenter omkring hvad der bruges til hvad (også omkring sikkerhedsaspektet:-)

Kommentar af thundergod d. 25. May 2004 | 3

Kommentar af kalsmose d. 15. May 2005 | 4

Kommentar af ciffer d. 23. Aug 2004 | 5

Giver en god intro til servlets/JSP. En uddybende artikel omkring disse emner ville være super :-)

Kommentar af radbyx d. 27. Jul 2004 | 6

Kommentar af jenniferw d. 06. Sep 2005 | 7

God generel introduktion.