Seminář Java

XI

Servlety, JSP

Co je servlet?

- Předkompilované java programy běžící na straně www serveru
- Standardní metoda metoda rozšiřování webových serverů
  o dynamické generování obsahu
- Java = různé platformy a servery (IIS, Tomcat, Apache, . . .)
- Ekvivalence CGI
- Jeden process
- Široké Java API
- Další technologie Javy (JDBC, EJB, . . .)

- javax.servlet.*;
- HttpServlet

Výhody servletů

- Perzistence mezi voláními – odezva (CGI, FastCGI)
- K dispozici celé Java API
  - JDBC (pooling)
  - JAXP
  - EJB
- Běh zajišťuje kontejner
- Kvalitní free kontejnery
- Různí dodavatelé – standardizace Java Servlet API 2.4

Servletové kontejnery

- Jako rozšíření www serveru
  - IIS
  - Apache/JServ
- S www serverem
  - Jetty
  - Tomcat
- Aplikační servery (J2EE)
  - JBoss
  - Sun ONE Server

Životní cyklus servletu

1. **INIT** – voláno při prvním uživateli, ne pro každý request
2. **SERVICE** – každý request, nový thread
3. **doGet, doPost, doXxx** – GET, POST
4. **DESTROY** – rušení instance, ne po každém requestu

Servlet

```
public class Servlet1 extends HttpServlet {

  public void init() throws ServletException {
  }

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
        throws ServletException, IOException {

    response.setContentType("text/html; charset=iso-8859-2");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Servlet1</title></head>");
    out.println("<body bgcolor=\"#ffffff\">");
    out.println("</body></html>");
  }

  public void destroy() {
  }
}
```

Zpracování dat

HttpServletRequest – „data od klienta"

- getParameter(name)
  - null pokud není
  - data od uživatele (name–název formulářového prvku)
- getParameterValues(name)
  - Pokud je jenom jedna položka, tak pole o 1 prvku
- getParameterNames()

HttpServletResponse – „co pošleme klientovi"

- setStatus
- setContentType
- addCokie
- sendRedirect
- PrintWriter out = response.getWriter();
- OutputStream os = response.getOutputStream();

Cookies

- Servlet pošle jednoduché jméno a hodnotu klientovi
- Klient si tyto hodnoty uloží
- Klient vrátí tyto data, pokud se shoduje adresa, doména apod. (dle nastavení cookie)

K čemu se to využívá?

- Identifikace uživatele během „session" (viz. další slajdy)
- Uložení informací o uživateli
- Nastavení stránky pro uživatele
- Cílená reklama

Problémy s cookies

Problémem je soukromí, nikoliv bezpečnost

- Server si může pamatovat Vaše akce
- Server může sdílet informace z cookies s dalšími servery
- „Špatné" weby ukládají citlivé informace do cookies

Doporučené zásady

- Pokud to není nezbytně nutné, nepoužívat
- Prohlížeč může mít zakázané cookies
- Neukládat citlivé informace

Cookies

Některé metody

- getMaxAge()/setMaxAge()
  - +hodnota, 0, -hodnota
- getComment()/setComment()
- getDomain()/setDomain()

Cookies

Nastavení cookie

```
Cookie c = new Cookie("myCookie","data");
c.setComment("Muj pokusny kolacek");
c.setMaxAge(10);
response.addCookie(c);
```

Cookies

Získání seznamu cookies

```
Cookie[] cookies = request.getCookies();
if (cookies !=null) {
  for (int i = 0; i < cookies.length; i++) {
    Cookie cookie = cookies[i];
    out.println("Cookie : "+cookie.getName()+
                " "+cookie.getValue());
  }
} else {
  out.println("<p>Zadna cookie neni k dispozici</p>");
}
```

Úkol

- Internetový obchod – nákupní košík
- Každý klient má svůj košík

Jak na to?

- Cookies
- URL-rewriting
- Skryté HTML formulářové prvky

Je potřeba vyšší API

Session

- Session objekt „žije" na serveru
- Pro usnadnění práce
- **HttpSession s = request.getSession(true);**
  - cookie/URL
  - klíč (SessionId)
  - vrátí předešlé/vytvoří novou
- Hashtable mechanismus

Session – metody

- HttpSession session = request.getSession(true);
- **session.setAttribute(name, value);**
- **session.getAttribute(name);**
- session.getAttributeNames();
- session.getCreationTime();
- **session.getId();**

Servlety – rekapitulace

- Programy na straně serveru
- Běží v servletovém kontejneru
- Ekvivalence CGI skriptům
- Běží na : IIS, Apache, Tomcat, . . .
- Specifikace
- Kvalitní „free" www servery (Tomcat, Jetty,. . .)

Servlety – co lze jednoduše a co ne?

- Číst formulářová data
- Číst hlavičky HTTP
- Zapisovat HTTP kódy a hlavičky
- Používat cookies
- Využívat session (uchování dat mezi dotazy uživatele)

- Špatně se píše HTML, XML apod.
- Co s tím??

JSP (Java Serve Pages)

Základní idea

- Použít HTML pro většinu stránky
- Označkovat servletový kód spec. tagy
- JSP stránku transformovat na servlet
- *Jaké jiné technologie tomu odpovídají?*

Co je tím myšleno?

- JSP
  <h2>Dobry den <%=request.getParameter("name")%></h2>
- URL
  http://localhost/hello.jsp?name=Pepiku
- Výsledek
  Dobry den Pepiku

JSP

Poznámka:

- Technicky nemůžou JSP dělat nic více, než může udělat servlet
- JSP nám pomáhají
  - Psát jednodušeji HTML
  - Lépe se orientovat v HTML stránky
  - Lze využít některé nástroje pro HTML (HomeSite)
- Oddělení prezentace od aplikační logiky
- Pořád je nutné umět programovat servlety
- JSP $\rightarrow$ servlet $\rightarrow$ klient
- Automatická kompilace
- Žádný spec. adresář
- Žádná definice v web.xml apod.
- Jak to funguje?

JSP – prvky

- Direktivy
  <%@ page import="com.xixao.*" %>
- Úsek programu
  <% int x = elements.size(); %>
- Deklarace <%! private int a = 0; %>
- Výraz V poli elements je <%=x%> položek
- Vestavěné značky
  <jsp:include page="dsdasda/xx.jsp"/>
- **Uživatelské značky**
  <mojeznacka:UkazDatum/>

JSP – příklad

```
<%@ page contentType="text/html; charset=iso-8859-2" %>
<%@ page import="java.util.*" %>
<html>
  <head>
    <title>IJA Test</title>
  </head>
  <body bgcolor="#ffffff">
    Datum <%=(new Date()).toString()%>
    <%
      for (int i=0; i<5; i++) {
    %>
      <hr width="20%" align="left"/>
      <h3><%=i%>. Ahoj</h3>
    <%
      }
    %>
  </body>
</html>
```

JSP – předdefinované proměnné

- HttpServletRequest – **request**
- HttpServletResponse – **response**
- PrintWriter – **out**
- HttpSession – **session**
- ServletContext – **application**
- PageContext – **pageContext**

JSP – příklad

```
<%@ page contentType="text/html; charset=iso-8859-2" %>
<html>
<head>
<title>IJA2</title>
</head>
<body bgcolor="#ffffff">
  <h1>Informace</h1>
  <p>
<%
    out.println("Session ID  : " +
                session.getId() + "<br>");
    out.println("Server info : " +
                application.getServerInfo() + "<br>");
%>
  </p>
 <b>Metoda pouzita pro dotaz : <%=request.getMethod()%></b>
</body>
</html>
```

JSP – JavaBeans komponenty

- Uživatelské třídy
- Musí mít konstruktor bez parametru
- Perzistentní prvky musí být přístupné přes getXXX/setXXX
- př: implementace nákupního košíku

Jak vypadá podpora v JSP

```
<jsp:useBean id="beanName"
    class="fully_qualified_classname" scope="scope">
    <jsp:setProperty .../>
 </jsp:useBean>
```

JSP – JavaBeans komponenty

```
<jsp:useBean id="kosik"
     class="com.ija.Kosik" scope="session"/>
```

Odpovídá

```
<% com.ija.Kosik kosik = new com.ija.Kosik(); %>
```

Proč tedy používat **<jsp:useBean/>**?

JSP – JavaBeans komponenty

<jsp:useBean/>

- Zjednodušuje zadávání parametrů z příchozího requestu
- Jednodušeji sdílí bean (parametr scope, application, . . .)

- Získání dat z beanu
  <jsp:getProperty name="name" property="property"/>

- Zadání dat do beanu
  <jsp:setProperty name="name" property="property"
  value="xx"/>

- Získání dat z requestu (konverze)
  <jsp:setProperty name="name" property="property"
  param="inputName"/>

JSP – Tagy

Co se musí udělat?

- Obslužná třída tagu
  - Musí implementovat javax.servlet.jsp.tagext.Tag
  - Obvykle se dědí z
    - TagSupport
    - BodyTagSupport
  - Překrývání metod
  - Stejný adr. na serveru jako servlety, beany
- Popisovač tagu
  - XML soubor, který popisuje tag (název, atributy, třída, . . .)
- JSP soubor
  - Import popisovače
  - Definice prefixu
  - Používat tagy

JSP – Obslužná třída tagu

```
package com.ija.tags

import javax.servlet.jsp.*;
import javax.servlet.jsp.tagexr.*;
import java.io.*;
import java.util.*;

public class MyDateTag extends TagSupport {
  public int doStartTag() {
    try {
      String date = (new Date()).toString();
      out.print(date);
    } catch(IOException e) {
      /* ...... */
    }
    return SKIP_BODY;
  }
}
```

JSP – Popisovač tagu

```
<?xml version="1.0"?>
<!DOCTYPE taglib ...>
<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.1</jspversion>

  <tag>
    <name>datum</name>
    <tagclass>com.ija.tags.MyDateTag</tagclass>
  </tag>

</taglib>
```

JSP – JSP

```
<%@ taglib uri="ija-taglib.tld" prefix="ija" %>

Pouziti : <prefix:JmenoTagu/>

 Dnesni datum je <b><ija:datum/></b>.
```

JSP – Vlastnosti uživ. tagu

- Tagy s atributy
  <ija:datum format="dd.mm.yyyy"/>
- Tagy, které čtou jejich obsah (zpracování, manipulace)
  <ija:datum> obsah </ija:datum>

- Mocný nástroj
- Usnadnění práce
- Listy, formuláře, přístup k DB . . .

JSP – Open Source Tag Libraries

http://jakarta.apache.org/taglibs/

- I18N – multijazyčné www
- Přístup k databázím
- XML, XSL transformace
- Práce s datem, časem
- Posílání emailů

JSP – Rekapitulace

- Snadné HTML
- Oddělení prezentace od aplikačního kódu
- Textové soubory, automatická kompilace
- Předdefinované objekty
- Uživatelské tagy
- OpenSource knihovny s uživ. tagy

Trendy ve www aplikacích

- Oddělení obsahu a vzhledu stránek
- Možnost výstupu v různých formátech (HTML, XHTML, XML, WAP, PDF)
- Jednoduchost programování
- Komponentní programování–znovupoužitelnost

- XML, XSL

XML x HTML

HTML

```
<h1>Ucty</h1>
<p>Pepa : 125,- Kc </p>
<p>Jana : 50,- Kc </p>
```

XML

```
<ucty>
   <osoba>
       <jmeno>Pepa</jmeno>
       <stav>125</stav>
   </osoba>
   <osoba>
       <jmeno>Jana</jmeno>
       <stav>50</stav>
   </osoba>
</ucty>
```

XML+XSL = HTML, XHTML, WAP, ...

Apache cocoon

- Založeno na komponentách
- XML, XSL
- Rourové zpracování
- Skriptovací jazyk, „continuations"
- OpenSource
- Založeno na servletu

- http://cocoon.apache.org
- Kdo to používá?

Rekapitulace

- Víme, co je servlet
- Jak se používá a programuje
- Víme, co je JSP
- Vztah servlet-JSP
- Výhody JSP
- Kam asi směřuje vývoj web aplikací

Kde lze pokračovat :

- http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html
- http://jakarta.apache.org/
- http://jetty.mortbay.com/jetty/index.html
- http://www.jboss.org

Co příště?

? J2ME ?