

Wprowadzenie do JSP

Marcin Apostoluk Tadeusz Pawlus Wojciech Walczak

Technologie Biznesu Elektronicznego, 7 marzec 2006



Wstęp

Porównanie z innymi technologiami

Opis Java Server Pages

Opis Java Bean

Opis Modelu 1 oraz Modelu 2

Opis serwera Jetty i Tomcat

Literatura

Plan prezentacji

- 1 Wstęp
- 2 Porównanie z innymi technologiami
- 3 Opis Java Server Pages
- 4 Opis Java Bean
- 5 Opis Modelu 1 oraz Modelu 2
- 6 Opis serwera Jetty i Tomcat
- 7 Literatura

Kto się tym zajmuje?

- Serwlety oraz JSP zostały stworzone w Sun Microsystems.
- Od wersji JSP 1.2 oraz Serwlety 2.3 rozwijane są w ramach JCP - Java Community Process
- Java Specification Request (JSR) 53 - opis JSP 1.2 oraz Serwletów 2.3
- JSR 152 opisuje JSP 2.0
- JSR 154 opisuje Serwlety 2.4 oraz 2.5
- JSR 245 opisuje JSP 2.1

Co to jest serwlet?

- Program pisany w języku Java
- Wykonywany na serwerach WWW lub serwerach aplikacji
- Pełni rolę warstwy pośredniczącej pomiędzy klientem oraz aplikacjami działającymi na serwerze

Kiedy warto używać serwletów

- Strona WWW generowana jest na podstawie informacji przesyłanych przez użytkownika
- Strona WWW jest tworzona na podstawie informacji, które często ulegają zmianie
- Strona WWW wykorzystuje zasoby pochodzące z korporacyjnej bazy wiedzy lub innych zasobów zgromadzonych na serwerze

Cykl życia serwletu

- Wywołanie metody `init`
- Każde wywołanie serwletu - nowy wątek
 - Metoda `service`
 - `SingleThreadModel`
- `doGet`, `doPost`, `doXxx` - zależy od otrzymanego żądania HTTP
- `destroy`

Przykładowy serwlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet{

    public void doGet( HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException{
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType = "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 " +
            "Transitional//EN">\n";
        out.println(docType + "<HTML>\n" + "<HEAD><TITLE>Witamy</TITLE></HEAD>\n" +
            "BODY BGCOLOR=#FDF5E6">\n" + "<H1>Witamy</H1>\n" + "</BODY></HTML>");
    }
}
```

Serwlety inne niż HTTP

- Serwlety można umieszczać na
 - Serwerach FTP
 - Serwerach poczty
 - Serwerach aplikacji
 - Serwerach HTTP
 - Wszystkie inne
- Mała popularność tych rozwiązań
- Serwery SIP - Session Initialization Protocol
`http://jcp.org/en/jsr/detail?id=116`

init, GET, POST,...

- `init` - tylko podczas inicjalizacji serwletu, gdy przetwarzany jest plik `web.xml`
- GET
- POST
- inne metody wg. W3C
 - OPTIONS
 - HEAD
 - PUT
 - DELETE
 - TRACE
 - CONNECT

Co to jest JSP?

- JSP = HTML + Java
- W jednym dokumencie kod HTML i zawartość generowana dynamicznie
- Kod Javy zwykle umieszczany pomiędzy znacznikami `<%` oraz `%>`

Przykład

```
<%@ page contentType='text/html'; charset=ISO-8859-2' %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Potwierdzenie zamówienia</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-2">
<LINK REL="STYLESHEET" HREF="JSP-Styles.css" TYPE="text/css">
</HEAD>
<BODY>
<H2>Potwierdzenie zamówienia</H2>
Dziękujemy za zamówienie <I><%= request.getParameter("title") %> </I>!
</BODY></HTML>
```

<http://localhost/StronaJSP.jsp?title=Nazwa+Ksiazki>

Różnice pomiędzy Serwletami i JSP

- Ogólnie można powiedzieć
- Serwlet = Java + HTML
- JSP = HTML + Java

strony JSP są przekształcane do postaci serwletów

Zalety rozdzielenia JSP i serwletów

- Oddzielenie warstwy logiki aplikacji od warstwy prezentacji
 - JSP - prezentacja
 - Serwlety - logika
- Strony JSP ułatwiają tworzenie oraz utrzymanie kodu HTML
- Tworząc strony JSP można wykorzystać standardowe narzędzia do tworzenia stron WWW
- Osoby nie znające Javy nie będą w stanie modyfikować kodu HTML
- Można podzielić zespół programistów

Co to jest aplikacja webowa

- Aplikacja uruchamiana na serwerze
- Komunikacja przez sieć (intranet lub Internet) za pomocą przeglądarki internetowej (cienki klient)

Cechy aplikacji webowych

- Łatwość zmian
 - Aplikacja w jednym miejscu
- Łatwość aktualizacji
- Ograniczone interfejsem przeglądarki
 - Brak Drag&Drop
 - Brak możliwości "rysowania po ekranie"
 - Rozwiązaniem jest dodanie skryptów wykonywanych po stronie klienta

RIA - Rich Internet Application

- Połączenie aplikacji webowej i standardowej aplikacji desktopowej
 - Używa standardowej przeglądarki internetowej, nie wymaga instalowania
 - Działa lokalnie w zabezpieczonym środowisku(`sandbox`)
 - Część obliczeń dokonywana jest przez klienta (gruby klient), reszta przez serwer aplikacji
- Termin wprowadzony przez Macromedia w marcu 2002 roku

Przykłady aplikacji webowych

- **GMail** <http://gmail.com/>
- **Google Maps** <http://maps.google.com/>
- **Wikipedia z mechanizmem edycji treści**
<http://pl.wikipedia.org/>
- **Biblioteka Główna Politechniki Wrocławskiej**
<http://www.bg.pwr.wroc.pl/>

Inne technologie

- Active Server Pages (ASP) – Microsoft
- PHP - Hypertext Preprocessor (PHP)
- ColdFusion – Macromedia
- JavaScript – Netscape
- Server-Side Includes (SSI)

Zalety JSP

- Szeroko obsługiwane – nie ogranicza się do określonego OS czy serwera HTTP
- Pełny dostęp do języka Java i technologii serwletów
- specyfikacja, nie produkt – wiele implementacji
- integralna część J2EE – szerokie zastosowanie

Składnia JSP(1)

- Kod JSP wewnątrz "<%>"
- Komentarz wewnątrz "<%- -" i "- -%>"
- Przykład:

adres: `http://nazwa_serwera/nazwastrony.jsp?osoba=Jan+Nowak`

```
<HTML>
```

```
<BODY>
```

```
Witaj <%= request.getParameter("osoba") %>!
```

```
</BODY>
```

```
</HTML>
```

Składnia JSP(2)

- Skrypt JSP
 - znaczniki HTML
 - elementy skryptu - tworzą kod Javy serwletu
 - dyrektywy - kontrola nad strukturą serwletu
 - akcje - określają komponenty (kontrola działania JSP)

Elementy skryptu(1)

- wyrażenie – przekazywane na wyjście

```
<%= wyrażenie %>
```

```
<jsp:expression> wyrażenie
```

```
</jsp:expression>
```

- Przykład:

Bierzący czas: `<%= new java.util.Date() %>`

Elementy skryptu(2)

- skryptlet – kod umieszczony wewnątrz metody service

```
<% kod %>
```

```
<jsp:scriptlet> kod </jsp:scriptlet>
```

- Przykład:

```
<%
```

```
String queryData = request.getQueryString();  
out.println("Dołączone dane typu GET:  
+ queryData);  
%>
```


Elementy skryptu(3)

- deklaracja – kod umieszczony w ciele klasy serwletu

```
<%! kod %>
```

```
<jsp:declaration> kod </jsp:declaration>
```

- Przykład:

```
<%! private int Licznik = 0; %>
```

Stronę wywołano od czasu uruchomienia serwera:

```
<%= ++Licznik %>
```

Dyrektywy(1)

```
<\%@ dyrektywa atrybut_1="wartosc_1"  
  atrybut_2="wartosc_2"  
  ...  
  atrybut_N="wartosc_N" %>
```

dyrektywy:

- page
- include
- taglib

Dyrektywy(2)

- Dyrektywa page – definiowanie atrybutów
- Przykłady:

```
<%@ page import="java.util.*" %>
```

```
<%@ page extends="pakiet.klasa" %>
```

```
<%@ page language="java" %>
```

Dyrektywy(3)

- Dyrektywa include – dołączanie plików w czasie przetwarzania strony JSP na serwlet
- Zawartość załączanego pliku jest traktowana jako tekst JSP `<%@ include file="względny URL" %>`

Dyrektywy(4)

- Dyrektywa taglib – biblioteka znaczników
- Przykład:

```
<%@ taglib url="/mojeznaczniki" prefix="moje" %>  
...  
<moje:mojedane> ... </moje:mojedane>
```

Akcje(1)

- `jsp:include` – **plik dołączany w momencie wywołania**
`<jsp:include page="wirtualny URL"
flush="true"/>`
- `jsp:useBean` – **znajduje lub buduje Java Bean**
`<jsp:useBean atr=wart*/> lub
<jsp:useBean atr=wart*> ... </jsp:useBean>`
- `jsp:setProperty` – **określa właściwości Java Bean**
`<jsp:setProperty atr=wart*/>`

Akcje(2)

- `jsp:getProperty` – **pobiera właściwość Java Bean**
`<jsp:getProperty name="nazwaWlasciwosci"
value="wartosc"/>`
- `jsp:forward` – **przekierowuje do innej strony**
`<jsp:forward page="wirtualny URL"/>`
- `jsp:plugin` – **tworzy znacznik OBJECT lub EMBED**
`<jsp:plugin attribute="wartosc"*> ...
</jsp:plugin>`

Akcje(3)

Zmienne predefiniowane:

- `request` - `HttpServletRequest`, dostęp do parametrów wywołania, typu wywołania, nagłówków HTTP
- `response` - `HttpServletResponse`, związana z odpowiedzią do klienta
- `out` - `PrintWriter`, wysyłanie danych do klienta
- `session` - `HttpSession`, związana z wywołaniem
- `application` - `ServletContext` otrzymany z `getServletConfig().getContext()`
- `config` - `ServletConfig` dla tej strony
- `pageContext` - `PageContext`
- `page` - **synonim** `this`

JavaBeans(1)

- Standardowy format budowy klas
 - Konstruktor domyślny
 - Wszystkie atrybuty niepubliczne
 - Dostęp do atrybutów poprzez metody `get/is` oraz `set`
 - Przykładowo
 - `int quantity`: `getQuantity()`, `setQuantity()`
 - `boolean empty`: `isEmpty()`, `setEmpty()`

JavaBeans(2)

- w stronach JSP: możliwość manipulowania wyłącznie składni XMLowej
- łatwo odczytywalna budowa klasy (na podstawie jej metod)
- wiele zastosowań
- często zawierają logikę biznesową

JavaBeans(3)

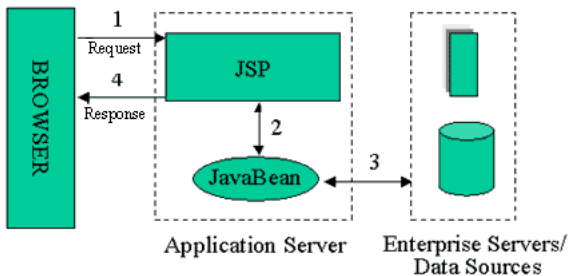
● Pobieranie wartości

```
<jsp:getProperty name="book1" property="title" />  
<\%= book1.getTitle() \%>
```

● Ustalanie wartości

```
<jsp:setProperty name="book1" property="title" value="new title" />  
<\%= book1.setTitle("new title"); \%>
```

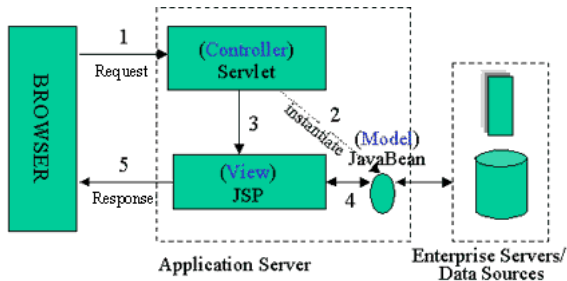
Model 1



Model 1

- Używany w prostych aplikacjach
- JSP - odbieranie żądań, prezentacja danych
- Dostęp do danych - JavaBeans
- W przypadku dużych aplikacji nadmierna ilość kodu odpowiedzialnego za odbieranie żądań

Model 2



Model 2

- Serwlet - funkcja kontrolera
- JSP - prezentacja danych, brak kodu logiki aplikacji
- Dostęp do danych - JavaBeans
- Serwlet buduje JavaBeans i przekazuje do stron JSP

Jetty(1)

- Serwer HTTP i kontener Serwletów
- `http://jetty.mortbay.org/jetty/`
- Jetty 6.0 BETA - najnowsza wersja
- Jetty 4.2.25 - używana, zgodność pluginów

Jetty(2)

- Tworzony pod nadzorem Mort Bay
`www.mortbay.com`
- Zgodność ze specyfikacją
- Open Source - licencja Apache 2.0
- Mały i efektywny:
 - jeden z najszybszych kontenerów serwletów
 - duża skalowalność - tysiące połączeń
 - optymalizowany od 1995 roku

Jetty(3)

- Pewny - użyty w wielu projektach
- Długa lista aplikacji wykorzystujących Jetty
`http://www.mortbay.com/mortbay/powered.html`
m. in. AliceBot, BugKill, JIRA, zintegrowany z Jeronimo, JBoss
- Embeddable(osadzalny) - zapewnia interfejs HTTP

Jetty(4)

- **Pluginy do środowiska Eclipse**

- <http://eclipse-plugins.2y.net>
- <http://www.eclipseplugincentral.com/>
- <http://www.eclipse.org/community/plugins.php>

- **Jetty Laucher**

- uruchamianie aplikacji web'owych z Eclipse'a
- debug'owanie aplikacji web'owych

Tomcat(1)

- Jeden z najpopularniejszych kontenerów serwletów
- Oficjalna, referencyjna implementacja serwletów/JSP
- <http://tomcat.apache.org/>
- Tomcat 5.5.15 - najnowsza stabilna wersja

Tomcat(2)

- Rozwijany w Apache Software Foundation
 - do wersji 5.0 rozwijany w ramach projektu Jakarta
- Darmowy, open-source – licencja Apache
- Zgodność ze specyfikacją
 - 5.5.x: Servlet 2.4, JSP 2.0

Tomcat(3)

- Wieloplatformowy – napisany w Java
- Zawiera serwer HTTP (stand-alone web server)
- Często stosowany z osobnym serwerem HTTP
- Wersja osadzalna

Tomcat(4)

- Najbardziej zgodny ze specyfikacją
- Najszybciej wprowadza specyfikację
- Pewny, niezawodny
- Gorzej udokumentowany od rozwiązań komercyjnych
- Trudniejsza konfiguracja, administracja
- Nieco wolniejszy

Tomcat(5)

Sysdeo Eclipse Tomcat Launcher plugin//

(<http://www.sysdeo.com/eclipse/tomcatplugin/>)

- Wersje:

- eclipse od 2.1.1
- tomcat od 3.3

Tomcat For Eclipse

(<http://openknows.free.fr/>)

Merve Tomcat Laucher

(<http://merve.sourceforge.net/>)

Literatura I



Marty Hall

More Servlets and JavaServer Pages.

Sun Microsystems Press/Prentice Hall PTR Book



Hans Bergsten

Java server pages.

O'Reilly



Marty Hall

Core Servlets and Java Server Pages.

A Sun Microsystems Press/Prentice Hall PTR Book

Literatura II



Lou Marco

EJB & JSP: Java On The Edge, Unlimited Edition.

M.&.T Publishing Ltd



Bruce Eckell

Thinking in Java 2ed.

Prentice-Hall



Marty Hall, Larry Brown

Java Servlets i Java Server Pages, wydanie II

Helion, 2006

Literatura III



<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1
World Wide Web Consortium



<http://www.stelmach.info/>
Krzysztof Stelmach - Kurs JavaServer Pages