pawel.rajba@gmail.com, http://kursy24.eu/

Servlets, JSP, JSTL

Plan wykładu

- Servlets
  - Charakterystyka
  - Główne metody
  - Obiekty request i response
- JSP
  - Dyrektywy
  - Elementy skryptowe
  - Elementy akcji
  - Generowanie treści
  - Obiekty niejawne
  - Obiekty request i response
  - Ciasteczka
  - Sesje
- JSTL
  - Przegląd znaczników

Servlets

- Krótka charakterystyka
  - Programy do generowania odpowiedzi HTTP
  - Jest to program w języku Java
  - Pozwala na tworzenie dynamicznych stron WWW
  - Kompilowane przy pierwszym żądaniu
  - Bardzo niskopoziomowe, rzadko bezpośrednio używane w tworzeniu serwisów internetowych
    - Dlatego nie będziemy im poświęcać więcej uwagi

Przykładowy servlet

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

public class HelloWorld extends HttpServlet
{
    public void doGet( HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```

Servlets

- Główne metody servletu
  - doGet
  - doPost
  - doPut
  - doDelete
  - init
  - destroy
- Ważne obiekty
  - request, response

Obiekty request, response

- request – reprezentuje żądanie
  - Daje dostęp do
    - Parametrów żądania, nagłówków żądania, ciastek
    - … i wielu innych składowych
- response – reprezentuje odpowiedź
  - Pozwala na
    - Przekierowanie odpowiedzi, ustawienie kodu odpowiedzi, dodanie ciastek
    - … i wiele innych operacji

JSP – wprowadzenie

- Technologia serwerowa, analogicznie do PHP
- JSP jest kompilowane
- Jako część JEE daje dostęp do platformy Java
- Obecnie podstawa do tworzenia stron w Javie
- Bardzo dojrzałe rozwiązanie

JSP – wprowadzenie

- Co może osadzić na stronie JSP?
  - Dyrektywy
  - Skrypty
  - Akcje
- Mamy dwa sposoby osadzania elementów:
  - typu JSP: <% coś %>
  - typu XML: <jsp:element>coś</jsp:element>

Dyrektywy

- Składnia
  - <%@ nazwa_dyrektywy %>
  - <jsp:directive.nazwa_dyrektywy />
    - Rodzaj składni często zależy od kontekstu
- Podstawowe dyrektywy
  - page – określa parametry strony
  - include – dołącza zewnętrzny plik
  - tablig – udostępnia zewnętrzną bibliotekę znaczników

Dyrektywy

- Dyrektywa page
  - Pozwala na ustawienie parametrów strony JSP
  - Wybrane atrybuty
    - contentType
      - typ zawartości (domyślnie text/html z ISO-8859-1)
    - errorPage
      - Strona wyświetlana w przypadku wyjątku
    - isErrorPage
      - Strona, która ma być wyświetlana przy pojawieniu się wyjątku musi mieć to ustawione na true (pozostałe strony mają na false)

Dyrektywy

- Dyrektywa page, c.d.
  - Wybrane atrybuty, c.d.
    - import
      - Wczytanie przestrzeni nazw (domyślnie: javax.Servlet.jsp.*, javax.Servlet.http.*, java.lang.*, javax.Servlet.*)
    - session
      - Określa, czy żądanie ma być częścią sesji (domyślnie true)
    - Uwaga: para atrybut/wartość może wystąpić raz
- Przykład
  - `<%@ page language="java„ contentType="text/html;charset=iso-8859-2" %>`
  - `<%@ page import="java.util.StringTokenizer" %>`

Dyrektywy

- Dyrektywa include
  - Dołącza zawartość zewnętrznego pliku
  - Wstawienie jest w momencie kompilacji
    - dołączenie jest statyczne
  - Atrybut file określa zasób do dołączenia
  - Przykład
    - `<%@ include file="naglowek.html" %>`

Dyrektywy

- Dyrektywa taglib
  - Udostępnia dodatkową bibliotekę znaczników
  - Atrybuty
    - uri – ścieżka do biblioteki
    - prefix – przestrzeń nazw dla znaczników
- Przykład
  - ```
    <%@ taglib
          uri="http://www.java.com/taglib/sample.jar"
          prefix="sample" %>
    ```
  - ```
    <sample:tagaction>Przykładzik</sample>
    ```

Elementy skryptowe

- Elementy skryptowe
  - Mamy 3 rodzaje elementów
    - Skryptlety
      <% kod %> <jsp:scriptlet>kod</jsp:scriptlet>
    - Wyrażenia
      <%= text %> <jsp:expression>text</jsp:expression>
    - Deklaracje
      <%! kod %> <jsp:declaration>kod</jsp:declaration>
      - Deklaracje nie generują żadnego „outputu”
      - Próba wywołania out.println spowoduje błąd kompilacji

Elementy akcji

- Osadzamy, aby wykonać jakąś akcję ☺
- Elementy akcji mogą być grupowane w biblioteki znaczników (tag libraries)
- Mamy do dyspozycji predefiniowane akcje
  - include, forward, plugin
- Możemy także tworzyć własny JavaBean i realizować do niego dostęp
  - useBean, setProperty, getProperty

Elementy akcji

- Podstawowa składnia
  - <jsp:nazwa_akcji atrybut="wartosc" />
- Dla wszystkich elementów akcji dostępne są atrybuty:
  - id – identyfikator
  - scope – określa czas życia elementu akcji:
    - page, request, session, application
      - różnica między page a request: po przekierowaniu przez np. akcję forward obiekt akcji page jest usuwany, a obiekt akcji request dalej dostępny

Elementy akcji

- Akcja include – wczytanie pliku
  - <jsp:include page="adres" flush="true" />
  - <jsp:include page="adres" flush="true">
    <jsp:param name="param" value="wartosc" />
    </jsp:include>
- Akcja forward - przekierowanie
  - <jsp:forward page="adres" />
  - <jsp:forward page="adres">
    <jsp:param name="param" value="wartosc" />
    </jsp:forward>

Elementy akcji

- Akcja plugin – osadzenie komponentu Java
  - Składnia

```
<jsp:plugin
        type="bean|applet"
        code="kod"
        codebase="kod_bazowy"
        [align="ustawienie"]
        [archive="lista_archiwow"]
        [height="wysokosc"]
        [hspace="odleglosc_pozioma"]
        [jreversion="wersja_JRE"]
        [name="nazwa_elementu"]
        [vspace="odleglosc_pionowa"]
        [width="szerokosc"]
        [nspluginurl="download_dla_Netscape"]
        [iepluginurl="download_dla_IE"]>
        [<jsp:params>
                [<jsp:param name="parametru "value="wartosc"/>]+
        </jsp:params>]
        [<jsp:fallback>tekst</jsp:fallback>]
</jsp:plugin>
```

Elementy akcji

- Akcja plugin, przykład:

```
<jsp:plugin type="applet"
     code="gierka.class"
    codebase="/aplety"
    width="200"
     height="200">
    <jsp:param name="param" value="val" />
    <jsp:fallback>
       Nie mozna uruchomic Java Plugin
     </jsp:fallback>
</jsp:plugin>
```

Elementy akcji

- Akcja JavaBeans
  - Najpierw krótkie przypomnienie jak tworzyć Bean'a
    - Konstruktor jest bezparametrowy
    - Dostęp do właściwości przez odpowiednie settery i gettery
      - public {Typ obiektu} get{Własciwosc} ()
      - public void set{Własciwosc} ({Typ obiektu})
      - public boolean is{Własciwosc} ()
    - ewentualnie do właściwości indeksowanych
      - public {Typ obiektu}[] get{Własciwosc} ()
      - public void set{Własciwosc} ({Typ obiektu}[])
      - public {Typ obiektu} get{Własciwosc} (int)
      - public void set{Własciwosc} (int, {Typ obiektu})

Elementy akcji

- Akcja JavaBeans
  - Atrybuty: id, scope, class, type, beanName
    - class – implementacja obiektu; domyślnie z beanName
    - type – typ zmiennej skryptowej; domyślnie brany z class
  - Obiekt jest wyszukiwany na podstawie id i scope
    - ewentualnie tworzona jest nowa instancja
  - Do ustawienia i pobrania wartości bean-a mamy
    - <jsp:setProperty name="" property="" value="" />
    - <jsp:getProperty name="" property="" />

Generowanie treści

- Wykorzystujemy elementy skryptowe
  - Wypisanie bieżącej daty na kilka sposobów:
    - `<% out.println( new java.util.Date() ); %>`
    - `<%= new java.util.Date() %>`
    - ```
      <jsp:scriptlet>
      out.println(new java.util.Date());
      </jsp:scriptlet>
      ```
    - ```
      <jsp:expression>
      new java.util.Date()
      </jsp:expression>
      ```

Przykłady

- data2.jsp
- data3.jsp
- akcja_include.jsp, tytul.jsp
- akcja_forward.jsp, data.jsp
- databean.jsp, ProstyBean.java
- formbean.jsp, formbeanconfirm.jsp, formbeanshow.jsp, UserInfoBean.java

Obiekty niejawne

- Ważne obiekty predefiniowane
  - request
    - typu javax.servlet.http.HttpServletRequest
  - response
    - typu javax.servlet. http.HttpServletResponse
  - session
    - typu javax.servlet. http.HttpSession
  - application
    - typu javax.servlet.ServletContext

Obiekty niejawne

- c.d.
  - out, typu javax.servlet.jsp.JspWriter
  - exception, typu java.lang.Throwable
    - dostępny tylko na stronach błedów
      - dostarcza informację o błędach
  - config, typu javax.servlet.ServletConfig
  - page, typu java.lang.Object
  - pageContext, typu javax.servlet.jsp.PageContext

Obiekt request

- Parametry żądania HTTP
  - request.getQueryString()
    - zwraca String
  - request.getMethod()
    - zwraca String
  - request.getParameter("imie")
    - zwraca String
  - request.getParameterValues("miasta")
    - zwraca String[]
  - request.getParameterMap()
    - zwraca Map<String,String[]>
  - request.getParameterNames()
    - zwraca Enumeration<String>

Obiekt request

- Nagłówki żądania HTTP
  - request.getHeader(String)
    - zwraca String
  - request.getHeaderNames()
    - zwraca Enumeration
  - request.getHeaders(String)
    - zwraca Enumeration
  - request.getLocale()
    - zwraca String (preferowane Accept-Language)
  - request.getLocales()
    - zwraca Enumeration (wszystkie Accept-Language)

Przykłady

- parameters.jsp
- headers.jsp

Obiekt response

- Dodanie nagłówków odpowiedzi
  - response.addHeader(String name, String value)
  - response.setHeader(String name, String value)
- Przykładowe nagłówki
  - Content-Type: text/html
  - Connection: keep-alive | close
  - Location: index2.html
  - Expires: <*data*>

Obiekt response

- Kody odpowiedzi
  - response.sendError(int sc)
  - response.sendError(int sc, String msg)
  - response.sendStatus(int sc)
  - response.sendStatus(int sc, String msg)
- Przykładowe kody odpowiedzi
  - Kod stanu — Stała w JSP
    - 200 — OK SC_OK
    - 204 No Content — SC_NO_CONTENT
    - 301 Moved Permanently — SC_MOVED_PERMANENTLY
    - 302 Moved Temporarily — SC_MOVED_TEMPORARILY
    - 401 Unauthorized — SC_UNAUTHORIZED
    - 403 Forbidden — SC_FORBIDDEN
    - 404 Not Found — SC_NOT FOUND
    - 500 Internal Server Error — SC_INTERNAL_SERVER_ERROR
    - 503 Service Unavailable — SC_SERVICE UNAVAILABLE

Obiekt response

- Kilka uwag
  - Do ustawienia nagłówka Content-type jest osobna funkcja, przykładowe użycie:
    - response.setContentType("text/xml");
  - Do sprawdzenia, czy dany nagłówek został wysłany jest metoda
    - boolean response.containsHeader(String name)
  - Dwa przykłady na przekierowanie strony
    - response.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY); response.setHeader("Location","http://localhost:8080/");
    - void response.sendRedirect("http://localhost:8080/");
      - Warto tutaj porównać do akcji <jsp:forward page="" />

Przykłady

- redirect.jsp, headers.jsp
- statushttp.jsp

Ciasteczka

- Obiekt Cookie
  - konstruktor
    - public Cookie(String name, String value)
  - metody
    - void Cookie.setMaxAge(int expire)
    - void Cookie.setDomain(String pattern)
    - void Cookie.setPath(String uri)
    - void Cookie.setSecure(boolean flag)
    - String Cookie.getName()
    - String Cookie.getValue()

Ciasteczka

- Tworzenie ciasteczka
  - void response.addCookie(Cookie cookie)
- Pobieranie ciasteczek
  - Cookie[] request.getCookies()

Przykład

- licznik.jsp

Sesje

- Identyfikator sesji jest przekazywany przez ciastko
  - Domyślnie nazywa się JSESSIONID
- Sesje są włączane domyślnie
  - Jeśli chcemy wyłączyc śledzenie sesji, musimy to zrobić jawnie poprzez dyrektywę
    - <%@ page session="false" %>
- Niejawnie tworzony jest obiekt session
  - Podstawowe metody:
    - Object getAttribute(String name)
    - void setAttribute(String name, Object value)
    - void removeAttribute(String name)
    - Enumeration<String> getAttributeNames()

Sesje

- Metody związane z cyklem życia sesji
  - int session.getMaxInactiveInterval()
    - pobiera czas, po jakim sesja wygaśnie
  - void session.setMaxInactiveInterval(int interval)
    - ustawienie czasu, po jakim sesja wygaśnie
  - long session.getCreationTime()
    - moment utworzenia sesji (liczba sekund od 1970-01-01)
  - long session.getLastAccessedTime()
    - moment ostatniego użycia sesji
  - void session.invalidate()
    - usuwa sesję

Przykłady

- odwiedziny.jsp
- confirmLogin.jsp, login.jsp, logout.jsp

JSTL

- JSTL to JavaServer Pages Standard Tag Library
- JSTL oferuje szereg pomocnych znaczników
  - Podstawowe znaczniki
  - Znaczniki formatujące
  - Znaczniki dla SQL
  - Znaczniki dla XML
  - Funkcja JSTL

JSTL

- Tutorial wraz z przykładami:
  - http://www.tutorialspoint.com/jsp/jstl_core_set_tag.htm
- Ciekawy przykład: JSTL + formularz
  - http://www.java2s.com/Code/Java/JSTL/JSTLFormUsingTextField.htm