JSP

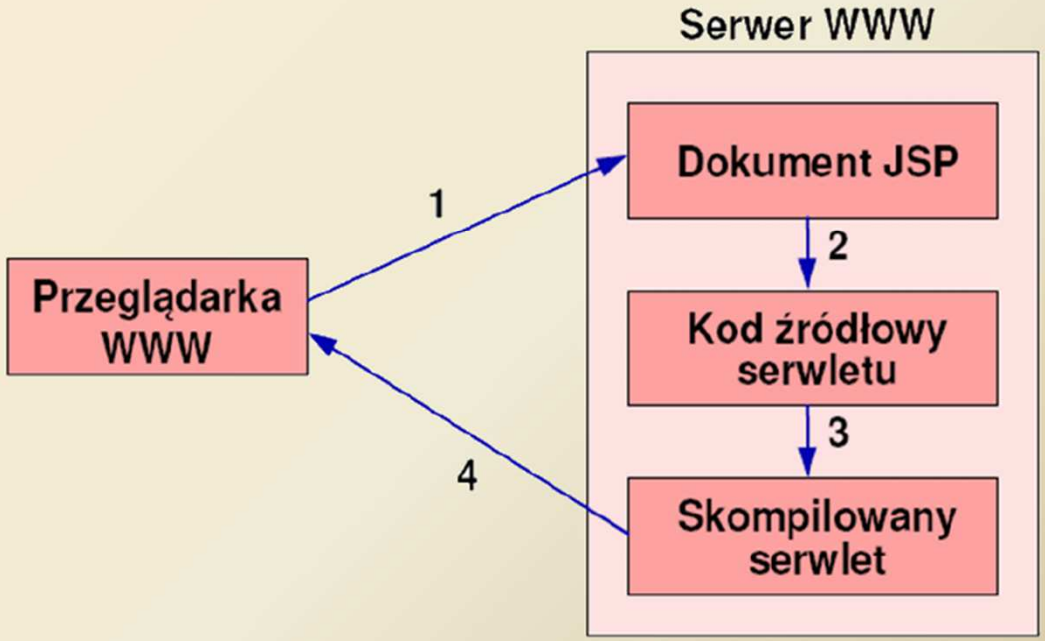
Co to jest JSP?

- Technologia Java Server Pages (JSP) to technologia szablonów umożliwiających łatwe łączenie statycznego kodu HTML lub XML z dynamiczną zawartościągenerowaną przez kod Java.
- Rozszerzenie technologii serwletów
- umożliwia osadzanie kodu Javy w dokumencie html
- Podstawowe narzędzie tworzenia warstwy prezentacji w architekturze Java EE

Skryptlet- kod osadzony w dokumencie html za pomocą znaczników jsp

Strony JSP

- Strona JSP jest mieszanką elementów oraz szkieletu
- Szkielet stanowi część strumienia wyjściowego
- Elementy zamieniane są na treść lub wykonują akcje
- Strona JSP kompilowana jest do kodu Javy i uruchamiana zgodnie ze specyfikacją Java Servlet



JSP-Cykl życia

Zalety i wady JSP

```
<html>
  <body>
    <%! String name = new String("Hello"); %>
    <%! public String getName() { return name; } %>
    Hello <b> <%= getName() %> </b>
  </body>
</html>
```

```
class servlet_xxxxxxx extends HttpServlet {
  String name = new String("Hello");
  public String getName() { return name; }
  public void doGet(HttpServletRequest request,
       HttpServletResponse response) {
    out.println("<html><body>");
    out.println("Hello <b>");
    out.println( getName() );
    out.println("</b>");
    out.println("</body></html>");
  }
}
```

Elementy stron JSP

- Skryptlety <% ... %>
- Komentarze <%-- ... --%> - inne będą widoczne w kodzie HTML
- Dyrektywy <%@ ... %> - page, taglib, include
- Deklaracje <%! ... %> - zmienne lub metody
- Wyrażenia <%= ... %> - wynik wstawiany od razu na stronę
- Akcje <jsp:... > - np. dołączenie kolejnych fragmentów strony

Dyrektywy

Kontrolują sposób translacji JSP do serwletu

Dyrektywy umieszczone są w znacznikach <%@%>

Dostępne są trzy dyrektywy:

- <%@ include %>: włączenie zewnętrznego pliku
- <%@ page %>: ustawienia strony
- <%@ taglib %>: wskazanie na bibliotekę znaczników

<%@page contentType="text/html" pageEncoding="UTF-8"%>

Deklaracje

- Pozwalają na deklarowanie metod i składowych serwletu wynikowego
- Mogą zawierać inicjalizację
- Wprowadzane przez znaczniki <%! %>

```
<%! int licznik = 0; %>
<%! int a, b, c; %>
<%! Array mojaTablica = new Array(); %>
```

Wyrażenia

- Znaczniki XML umożliwiające wartościowanie wyrażenia
- Wartość konwertowana do łańcucha znaków i włączana
- do wynikowego kodu HTML lub XML
- Wprowadzane przez znaczniki <%= %>

```
<%! Calendar today = new GregorianCalendar(); %>
<P>Dzisiejsza data to
<%= today.get(Calendar.DAY_OF_MONTH) %>,
<%= today.get(Calendar.MONTH) %>, roku
<%= today.get(Calendar.YEAR) %>
</p>
```

Skryptlety

- Znaczniki XML umożliwiające osadzanie kodu Java
- Mogą generować kod HTML lub XML za pomocą
- predefiniowanego obiektu out
- Wprowadzane przez znaczniki <% %>

```
<% Calendar dzis = new GregorianCalendar();
int godzina = dzis.get(Calendar.HOUR_OF_DAY);
out.println("<B>jest godzina " + godzina + "</B>");
%>
```

Znaczniki jsp - akcje

- Znaczniki XML wywołujące akcje serwera aplikacji
- Dostępne akcje
  - <jsp:include>: włączenie zewnętrznego kodu
  - <jsp:forward>: przekazanie sterowania
  - <jsp:param>: zdefiniowanie parametru
  - <jsp:plugin>: obsługa apletów Java
  - <jsp:fallback>: gdy klient nie obsługuje apletów
  - i znaczniki do obsługi komponentów JavaBean

Strona JSP a dokument JSP

- Strony JSP nie są poprawnymi dokumentami XML
- Standard JSP 1.2 wprowadza tzw. dokumenty JSP
- znacznik <jsp:root xmlns:jsp=adres>
- skryptlety, deklaracje i wyrażenia posiadająodpowiadające im znaczniki XML:
  - *<jsp:directive>*
  - *<jsp:declaration>*
  - *<jsp:expression>*
  - *<jsp:scriptlet>*
  - *<jsp:text>*

Scope

- scope - zasięg zmiennych przekazywanych między kolejnymi cyklami request-response

page, request, session, application

Scope-omówienie

- Page
  - Podstawowy, domyślny scope. Oznacza, że obiekt z nim powiązany będzie istniał tylko w obrębie danej instancji strony
- Request
  - Obiekt istnieje w obrębie jednego zapytania. Jedno zapytanie może być obsługiwane przez wiele stron
- Session
  - Obiekt istnieje w obrębie całej sesji i jest dostępny z poziomu różnych stron
- Application
  - Scope globalny dla całej aplikacji

JSTL-co to jest?

- **JSTL (Java Standard Tag Library)** to standardowa biblioteka znacznikow Javy ulatwajaca pisanie stron JSP, tj. sprawiajaca ze strona JSP staje się „bardziej czysta”
- Korzystajac z JSTL-a bardzo czesto korzystamy rownież z Jezyka Wyrazeń (EL - Expression Language), który to wraz z szablonami był swego czasu alternatywą dla technologii JSP.

JSTL - znaczniki

- Podstawowe
  - *<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>*
- *Formatujące*
  - *<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>*
- *XML*
  - *<%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %>*
- *SQL*
  - *<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>*
- *Funkcje*
  - *<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>*

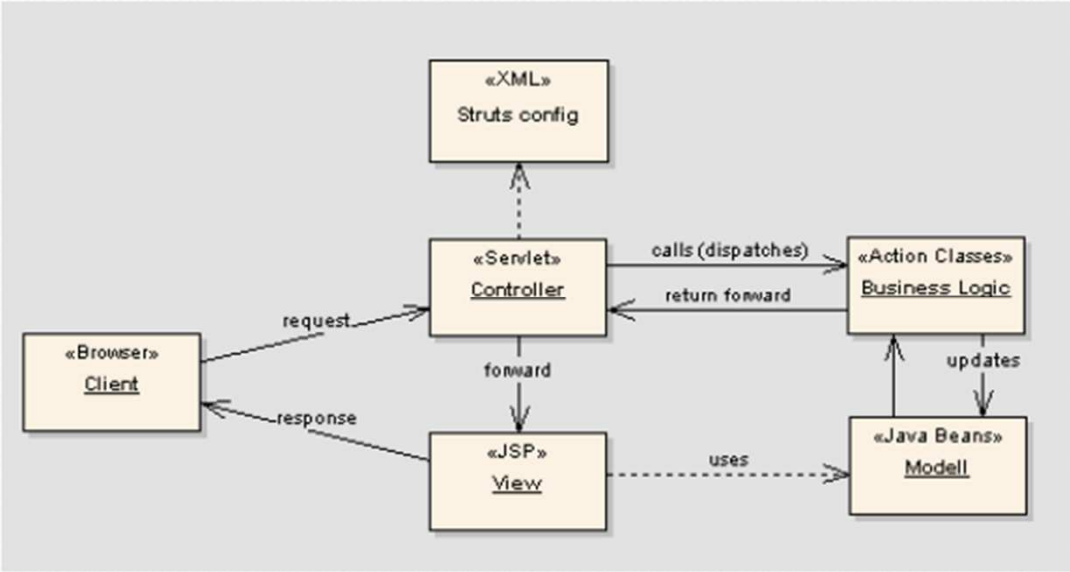
Obiekty predefiniowane

Obiekt	Typ	Zakres	Najczęściej używane metody
request	pochodna *javax.servlet.ServletRequest*	Request	getAttribute, getParameter, getParameterNames, getParameterValues, setAttribute
resposne	pochodna *javax.servlet.ServletResponse*	Page	*Standardowo nie używana przez autorów stron JSP*
pageContext	*javax.servlet.jsp.PageContext*	Page	findAttribute, getAttribute, getAttributesScope, getAttributeNamesInScope, setAttribute
session	*javax.servlet.http.HttpSession*	Session	getAttribute, getId, setAttribute
application	*javax.servlet.ServletContext*	Application	getAttribute, getMimeType, getRealPath, setAttribute
out	*javax.servlet.jsp.JspWriter*	Page	clear, clearBuffer, flush, getBufferSize, getRemaining
config	*javax.servlet.ServletConfig*	Page	getInitParameter, getInitParameterNames
page	*java.lang.Object*	Page	*Standardowo nie używana przez autorów stron JSP*
exception	*java. lang. Throwable*	Page	getMessage, getLocalizedMessage, printStackTrace, toString

Expression Language

- Uproszczony sposób użycia tagów
- Na przykład zamiast:
  - `<%= pageContext.findAttribute("a") %>`
  - `<%= request.getParameter("p") %>`
- Można napisać:
  - `${a}`
  - `${param.p}`

MVC – Model View Controller



MVC przebieg cyklu request-response

1. Przeglądarka wysyła żądanie. Aplikacja jest tak skonfigurowana, że każde żądanie jest kierowane do serwletu – kontrolera.
2. Serwlet – kontroler analizuje żądanie i tworzy wymagane przez żądany widok obiekty klas zewnętrznych. Interakcja kontrolera z modelem może pociągać za sobą interakcję z bazą danych.
3. Serwlet przekazuje sterowanie do odpowiedniego widoku - strony JSP.
4. JSP pobiera dane z obiektów modelu przygotowanych przez kontroler. Obiekty te mogą udostępniać dane pobrane z bazy danych.
5. JSP generuje wynikowy dokument HTML przesyłany do przeglądarki.