

Wykład
JSP – Obiekty bezpośrednie,
Akcje JSP,
Obsługa błędów,
Tworzenie bibliotek znaczników

Zaawansowane programowanie w Javie

Dr inż. Damian Raczyński

Obiekty bezpośrednie



Zbiór predefiniowanych obiektów, gotowych do użycia:

- out:	JspWriter	Omówione wcześniej lub przy okazji serwletów
- request:	HttpRequest	
- response:	HttpResponse	
- session:	HttpSession	
- config:	ServletConfig	
- application:	ServletContext	Nowe
- pageContext:	Obiekt specjalny JSPpage	
- page:	JSPpage	

2

pageContext



pageContext może zostać wykorzystany do:

- ustawienia lub pobrania atrybutu z zasięgów: strony, żądania, sesji, aplikacji.
- dostępu do obiektów publicznych,
- uzyskania obiektu JspWriter,
- zarządzania sesją,
- przekierowania forward/include bieżącego żądania,
- obsługi wyjątku errorpage.

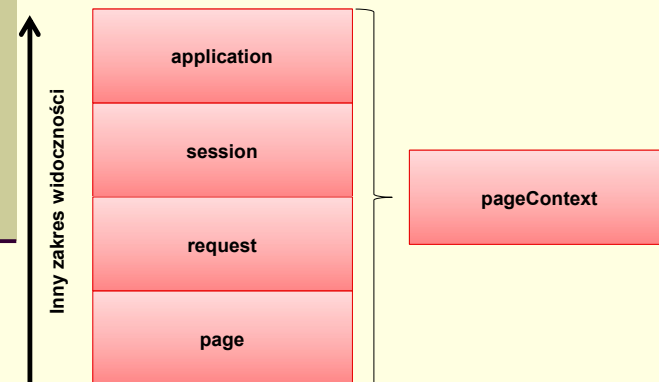
W JSP zasięg strony jest domyślny.

3

pageContext



obiekt pageContext - zakresy widoczności



4

pageContext



Wybrane metody dostępu do atrybutów:

typ	metoda	opis
Object	<code>findAttribute(String)</code>	szuka atrybutu w zasięgu widoczności strony/żądania/... lub zwraca null
Object	<code>getAttribute(String)</code>	zwraca atrybut strony/null
Object	<code>getAttribute(String, int scope)</code>	zwraca obiekt z określonego zakresu widoczności
Enumeration	<code>getAttributeNamesInScope(int scope)</code>	nazwy atrybutów z określonego zasięgu widoczności
int	<code>getAttributesScope(String)</code>	zwraca zasięg widoczności atrybutu
void	<code>removeAttribute(String)</code>	usuwa atrybut ZE WSZYSTKICH ZAKRESÓW WIDOCZNOŚCI
void	<code>setAttribute(string, Object)</code>	ustawia atrybut strony
void	<code>setAttribute(string, object, int scope)</code>	ustawia atrybut w określonym zakresie widoczności

pageContext



```
<%
JspWriter x = pageContext.getOut();
HttpSession ses= pageContext.getSession();
ses.setAttribute("attr", "SESJA");
pageContext.getServletContext().setAttribute("attr", "KONTEKST");
pageContext.getRequest().setAttribute("attr", "REQUEST");
pageContext.setAttribute("attr", "PAGE", pageContext.PAGE_SCOPE);
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
x.println("<br>" + pageContext.getAttribute("attr") + "</br>");
pageContext.removeAttribute("attr", pageContext.PAGE_SCOPE);
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
pageContext.removeAttribute("attr", pageContext.REQUEST_SCOPE);
x.println("<br>" + pageContext.getAttribute("attr") + "</br>");
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
x.println("<br>" + pageContext.getAttribute("attr",
pageContext.APPLICATION_SCOPE));
%>
```

pageContext



```
<%
JspWriter x = pageContext.getOut();
HttpSession ses= pageContext.getSession();
ses.setAttribute("attr", "SESJA");
pageContext.getServletContext().setAttribute("attr", "KONTEKST");
pageContext.getRequest().setAttribute("attr", "REQUEST");
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
x.println("<br>" + pageContext.getAttribute("attr") + "</br>");
pageContext.removeAttribute("attr", pageContext.PAGE_SCOPE);
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
pageContext.removeAttribute("attr", pageContext.REQUEST_SCOPE);
x.println("<br>" + pageContext.getAttribute("attr") + "</br>");
x.println("<br>" + pageContext.findAttribute("attr") + "</br>");
x.println("<br>" + pageContext.getAttribute("attr",
pageContext.APPLICATION_SCOPE));
%>
```

PAGE

PAGE

REQUEST

null

SESJA

KONTEKST

pageContext



Wybrane pozostałe metody:

metoda	opis
<code>forward(String relativeURL)</code>	przekierowanie forward
<code>include(String relativeURL)</code>	perzekierowanie include
<code>getException()</code>	bieżąca wartość obiektu wyjątku
<code>getRequest(), getResponse()</code>	zwraca obiekty żądania/odpowiedzi
<code>getServletConfig(), getServletContext(), getServletSession()</code>	zwraca obiekty konfiguracji/kontekstu/sesji

pageContext



```
index.jsp:
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<body>
<form action="2.jsp">
<input type="text" name="nazwa">
<input type="submit" value="OK"><br/>
</form>
</body>
</html>
```

9

pageContext



```
2.jsp:
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<body>
<%
String name=request.getParameter("nazwa");
out.print("Witamy "+name);
pageContext.setAttribute("uzytkownik",name,PageContext.SESSION_SCOPE);
%>
<a href="3.jsp">KOLEJNA</a>
</body>
</html>
```

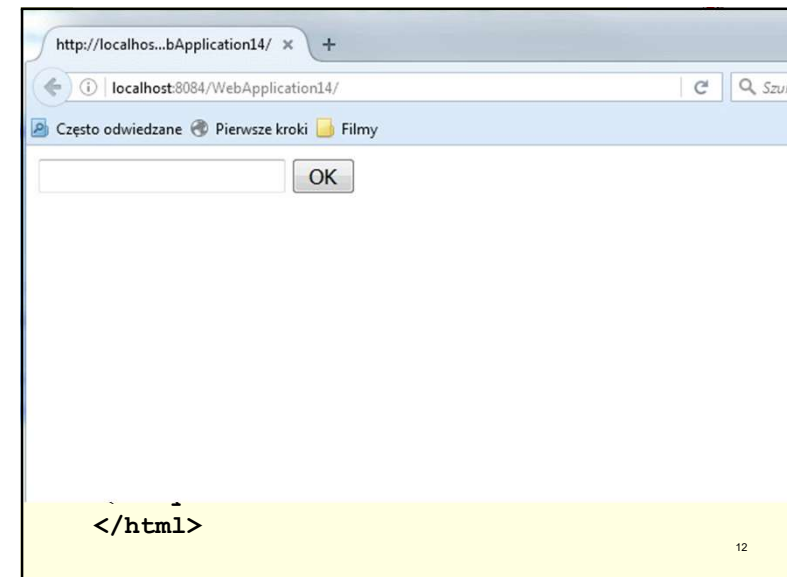
10

pageContext



```
3.jsp:
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<body>
<%
String
name=(String)pageContext.getAttribute("uzytkownik",PageContext.SESSION_SCOPE);
out.print("WITAMY po raz drugi "+name);
%>
</body>
</html>
```

11

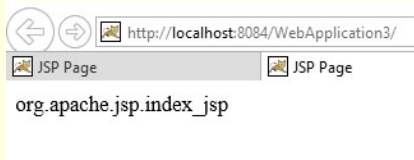


12

page

page obiekt bezpośredni reprezentujący instancję serwletu (jednoznaczny ze słówkiem this wewnątrz serwletu).

```
<%
    String x= page.getClass().getName();
%>
<%=x%>
```



13

Akcje JSP

Akcje mają za zadanie wykonać zdefiniowane zadanie w **CZASIE PRZETWARZANIA ŻĄDANIA** (dyrektywy działają w fazie tłumaczenia). najczęściej używane:

akcja	opis
jsp:forward	przekierowuje żądanie do innego elementu serwera
jsp:include	wstawia odpowiedź z innego źródła
jsp:element	definiuje element XML
jsp:getProperty	pobiera wartość właściwości z ziarna
jsp:useBean	tworzy obiekt ziarna
jsp:plugin	osadza inne elementy strony (np. aplety)
jsp:param	ustawia wartość parametru
jsp:setProperty	ustawia wartość właściwości ziarna
jsp:attribute	definiuje atrybuty elementu XML dynamicznie
jsp:body	definiuje zawartość elementu XML

Akcje JSP

<jsp:include>

wstawia stronę o podanej lokalizacji do strony, w której znacznik został wywołany

składnia:

```
<jsp:include page=„{adres}” flush=„true”/>
```

Dyrektywa include	akcja include
w czasie tłumaczenia	w czasie uruchomienia
Kopiuje wstawiany plik	Odwołuje się do wstawianego pliku
Dla zawartości statycznych	Dla zawartości dynamicznych
Nie może przekazywać parametrów	Może przekazywać parametry

Akcje JSP

Przekazywanie parametrów do wstawianej strony:

```
<jsp:include page=„{strona}” flush=„true”>
    <jsp:param name=„{nazwaP1}” value=„{wartoscP1}” />
</jsp:include>
```

16

```
index.jsp:
<body>
<%
request.setAttribute("attr", "Parametr ze strony glownej");
%>
<jsp:include page="second.jsp">
    <jsp:param name="jeden" value="red"/>
    <jsp:param name="dwa" value="green"/>
</jsp:include>
</body>
```

```
second.jsp:
<body>
    <h2> DRUGA STRONA </h2>
    <div style="border:1px solid black; background-color:
    <%=request.getParameter("jeden") %>;
    color: <%=request.getParameter("dwa") %>;
    width: 50%">
    <%= request.getAttribute("attr") %>
    </div>
</body>
```

```
index.jsp:
<body>
<%
request.setAttribute("attr", "Parametr ze strony glownej");
%>
<jsp:include page="second.jsp">
```

http://localhost:8084/WebApplication3/

JSP Page x JSP Page Apache Tomcat/8.... Apache Tomcat/8..

DRUGA STRONA

Parametr ze strony glownej

```

        color: <%=request.getParameter("dwa") %>;
        width: 50%">
        <%= request.getAttribute("attr") %>
    </div>
</body>
```

Akcje JSP

`<jsp:forward>`
przekierowuje żądanie użytkownika do innej strony

składnia:

`<jsp:forward page="{strona_do_przejscia}"/>`

Przekazywanie parametrów do przekierowanej strony:

```
<jsp:forward page="{strona}" flush="{true}"
    <jsp:param name="{nazwaP1}" value="{wartoscP1}" />
</jsp:forward>
```

19

```
index.jsp:
<body>
<h1>STRONA 1</h1>
<%
request.setAttribute("attr", "Parametr ze strony glownej");
%>
<jsp:forward page="second.jsp">
    <jsp:param name="op1" value="10"/>
    <jsp:param name="op2" value="20"/>
</jsp:forward>
<h1>Koniec strony 1</h1>
</body>
```

```
second.jsp:
<body>
    <h2> DRUGA STRONA </h2>
    <div>
        <%= request.getAttribute("attr") %>
    <%= request.getParameter("op1")+request.getParameter("op2") %>
    </div>
    <h2>Koniec strona 2</h2>
</body>
```



Akcje JSP



`<jsp:include>`
przekierowuje żądanie użytkownika do innej strony z powrotem do strony bieżącej (uzupełniana zostaje odpowiedź a nie tworzona na nowo)



składnia:

`<jsp:include page="{strona_do_przejscia}" />`

Przekazywanie parametrów do przekierowanej strony:

```
<jsp:include page="{strona}" flush="true">
  <jsp:param name="{nazwaP1}" value="{wartoscP1}" />
</jsp:include>
```

22

```
index.jsp:
<body>
<h1>STRONA 1</h1>
<%
  request.setAttribute("attr", "Parametr ze strony glownej");
%>
  <jsp:include page="second.jsp">
    <jsp:param name="op1" value="10"/>
    <jsp:param name="op2" value="20"/>
  </jsp:include>
  <h1>Koniec strony 1</h1>
</body>
```

```
second.jsp:
<body> <h2> DRUGA STRONA </h2><div>
<%= request.getAttribute("attr") %>
<%
  int x=Integer.parseInt(request.getParameter("op1"));
  int y=Integer.parseInt(request.getParameter("op2"));
  out.println("<hr>wynik="+ (x+y) + "<hr>");
%>
  </div> <h2>Koniec strona 2</h2>
</body>
```

```
index.jsp:
<h1>STRONA 1
<%
  request.setAttribute("attr", "Parametr ze strony glownej");
%>
  <jsp:include page="second.jsp">
    <jsp:param name="op1" value="10"/>
    <jsp:param name="op2" value="20"/>
  </jsp:include>
  <h1>Koniec strony 1
</body>
```

```
second.jsp:
<body> <h2> DRUGA STRONA </h2><div>
<%= request.getAttribute("attr") %>
<%
  int x=Integer.parseInt(request.getParameter("op1"));
  int y=Integer.parseInt(request.getParameter("op2"));
  out.println("<hr>wynik="+ (x+y) + "<hr>");
%>
  </div> <h2>Koniec strona 2</h2>
</body>
```

Akcje JSP



`<jsp:element>`, `<jsp:attribute>`, `<jsp:body>`
Elementy umożliwiają zdefiniowanie wartości znacznika (XML/HTML)

składnia:

```
<jsp:element name="nazwa">
  <jsp:attribute name="nazwa_atrybutu"
    wartość atrybutu
  </jsp:attribute>
  <jsp:body>
    wartość znacznika "nazwa" - to co się
    znajduje pomiędzy znacznikiem
    otwierającym a zamykającym
  </jsp:body>
</jsp:element>
```

25

```
index.jsp:
<body>
<%
  String serv=request.getServerName();
  String port=String.valueOf(request.getServerPort());
  String app= request.getContextPath();
  String link="/second.jsp";
  String tekst="Link do strony 2";
%>
<jsp:element name="a">
  <jsp:attribute name="href">
    <%= "http://" + serv + ":" + port + app + link %>
  </jsp:attribute>
  <jsp:body>
    <%= tekst %>
  </jsp:body>
</jsp:element>
</body>
```

26

index.jsp:

```
<body>
<!-- ... -->
</body>

WebApplication3 X
4 <!DOCTYPE html>
5 <html>
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8     <title>JSP Page</title>
9   </head>
10  <body>
11
12
13    <a href="http://localhost:8084/WebApplication3/second.jsp">
14      Link do strony 2
15    </a>
16  </body>
17 </html>
18
```

27