

Java technológiák – 5. előadás JSP lapok.

ANTAL Margit

Sapientia - EMTE

2009

- JSP alapfogalmak
- JSP szkriptelemek
- JSP standard tagok
- Expression Language

Hello servlet részlet (1)

```
import java.io*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorldServlet extends HttpServlet {
    private static final String DEFNAME ="Vilag";

    protected void generateResponse(...){
    }

    protected void doGet(...){
        generateResponse(request, response);
    }

    protected void doPost(...){
        generateResponse(request, response);
    }
}
```

Hello szervlet részlet (2)

```
protected void generateResponse
    (HttpServletRequest request,
     HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    String name = request.getParameter("name");
    if( name == null || name.length() == 0 )
        name = DEFNAME;
    out.println("Hello, "+name+"!");
    out.close();
}
```

Hello servlet konfigurálás

```
<servlet-mapping>  
  <servlet-name>HelloWorldServlet</servlet-name>  
  <url-pattern>/hello.view</url-pattern>  
</servlet-mapping>
```

```
http://localhost:8080/app/hello.view?name=Kati
```

Hello JSP lap

```
<%! private static final String
        DEFNAME = "vilag"; %>
<html>
  <head>
    <title>JSP Page</title>
  </head>
  <body>
    <%
      String name = request.getParameter("name");
      if( name == null || name.length() == 0 )
        name = DEFNAME;
    %>
    <h2>Hello, <%= name %> !</h2>
  </body>
</html>
```

```
void jspInit()
```

```
void _jspService(HttpServletRequest, HttpServletResponse)
```

```
void jspDestroy()
```


- 1 JSP lap átalakítása szervletté: `foo.jsp` \rightarrow `foo_jsp.java`, minden módosítás után
- 2 A szervlet fordítása: `foo_jsp.java` \rightarrow `foo_jsp.class`
- 3 A `foo_jsp.class` fájl betöltése
- 4 A `foo_jsp` osztály példányosítása
- 5 A `jspInit` metódus meghívása a létrehozott példányra
- 6 HTTP kérések kiszolgálása: `_jspService` metódus hívásával
- 7 A `jspDestroy` metódus hívása a komponens eltávolításakor

Szkriptelem	Példa
Megjegyzés	<code>< % -- <i>megjegyzes</i> -- % ></code>
Direktíva	<code>< %@ <i>direktiva</i> % ></code>
Deklaráció	<code>< %! <i>deklaracio</i> % ></code>
Szkriptlet	<code>< % <i>szkriptlet</i> % ></code>
Kifejezés	<code>< % = <i>kifejezes</i> % ></code>

- **HTML megjegyzés:** `<! -- HTML megjegyzes -- >`, része lesz a HTTP válasznak, tehát a kiszolgáló ezt is elküldi a böngészőnek
- **JSP megjegyzés:** `<% -- JSP megjegyzes -- %>`, a JSP megjegyzés csak a JSP lapon látható, ez nem kerül bele a lapból előállított szervlet osztályba, illetve a HTTP válaszbba sem
- **Java megjegyzés:** `<%/* Java megjegyzes /* %>`, a Java megjegyzés belekerül a JSP lapból előállított szervlet osztály kódjába, de nem kerül bele a HTTP válaszbba

- **page**: pl. Java import utasítások

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

- **include**: egy másik komponens tartalmának beszúrása

```
<%@ include file = "banner.jsp" %>
```

- **taglib**: elemkönyvtárak használata

```
<%@ taglib uri = "tlds/taglib.tld" prefix =  
"mytag" %>
```

Szintaxis

```
< %! deklaracio % >
```

Példák:

1. `<%! int counter = 0; %>`

2. `<%!
 public void jspInit(){
 //
 }
%>`

3. `<%!
 public void jspDestroy(){
 //
 }
%>`

Szkriptletek

```
<% for( int i=0; i<10; ++i){%>  
  
    <h2>Hello World!</h2>  
  
<% }%>
```

Kifejezések

$\langle \% = a\% \rangle, \langle \% = b\% \rangle$

$\langle \% = a + b + 10\% \rangle$

Objektum	Típus
request	javax.servlet.http.HttpServletRequest
response	javax.servlet.http.HttpServletResponse
out	javax.servlet.jsp.JspWriter
session	javax.servlet.http.HttpSession
application	javax.servlet.ServletContext
config	javax.servlet.HttpServletConfig
pageContext	javax.servlet.jsp.PageContext
page	java.lang.Object
exception	java.lang.Throwable

A page direktíva

- language
- extends
- buffer
- autoFlush
- session
- import
- isThreadSafe
- info
- contentType
- pageEncoding
- isELIgnored
- errorPage
- isErrorPage

Az include direktíva

```
<%@ include file="header.jsp" %>
```

Fontos!!!

A beillesztés még szervletté alakítás előtt történik.

- `<jsp:prefix .../>`
- `<jsp:prefix ...>`
 ...
 `</jsp:prefix>`

Jellemzők:

- nincs publikus attribútum,
- attribútumok kezelése: get/set metódusokkal,
- van paraméter nélkül hívható konstruktor,
- szerializálható.

```
<jsp:useBean.../>  
<jsp:setProperty.../>  
<jsp:getProperty.../>  
<jsp:include.../>  
<jsp:param.../>  
<jsp:forward.../>
```

A useBean tag

```
<jsp:useBean  
  id="beanName"  
  scope="page|request|session|application"  
  class="className"  
>
```

```
<jsp:useBean id="course" scope="request"
              class="model.Course">
  <%
    course.setName (request.getParameter("name"));
    course.setPrice( Integer.parseInt
                      (request.getParameter("price")))
  );
  %>
</jsp:useBean>
```

A setProperty tag

```
<jsp:setProperty name="beanName" propertyexpression />
```

- `property="*"`
- `property="propertyName"`
- `property="propertyName" param="paramName"`
- `property="propertyName" value="propertyValue"`

setProperty példa(2)

JSP standard elem

```
<jsp:setProperty name="course" property="name"/>
```

Ekvivalens Java kód

```
course.setName(request.getParameter("name"));
```

setProperty példa(2)

JSP standard elem

```
<jsp:setProperty name="course" property="name"  
                param="nev" />
```

Ekvivalens Java kód

```
course.setName(request.getParameter("nev"));
```

setProperty példa(3)

JSP standard elem

```
<jsp:setProperty
  name="course"
  property="name"
  value='<%= metodus() %>'
/>
```

Ekvivalens Java kód

```
course.setName(metodus());
```

setProperty példa(4)

setProperty

```
<jsp:setProperty name="course" property="*" />
```

A getProperty tag

- Szintaxis:

```
<jsp:getProperty name="beanName"  
                property="propertyName"/>
```

- Használat:

```
<jsp:getProperty name="course"  
                property="price"/>
```

- Ekvivalens Java kód:

```
out.print(course.getPrice());
```

Az include tag

beagyazo.jsp

```
<%! String name="Hello"; %>  
<jsp:include page="beagyazott.jsp" />
```

beagyazott.jsp

```
<% out.print(name); %>
```

A param tag

Beágyazó

```
<jsp:include page="beagyazott.jsp">  
  <jsp:param name="username" value="jojo"/>  
</jsp:include>
```

Beágyazott

A felhasználó neve:

```
<%= request.getParameter("username") %>
```

Az forward tag

```
<jsp:forward page="{pageURL" | "<%= expression %>"}" >  
  <jsp:param  
    name="parameterName"  
    value="{parameterValue | <%= expression %>}"  
  />  
</jsp:forward>
```

```
<jsp:forward page="masik.jsp">  
  <jsp:param name="username" value="jojo" />  
</jsp:forward>
```


Mikor használjuk?

- akcióelemben valamely attribútum megadásakor:

```
<jsp:include page='${URL_to_page}'/>
```

- HTML szövegben:

```
<H2>Hello, ${param.name}</H2>
```

- `pageContext`
- `pageScope (String - Object)`
- `requestScope (String - Object)`
- `sessionScope (String - Object)`
- `applicationScope (String - Object)`
- `param (String - String)`
- `paramValues (String - String[])`
- `header (String - String)`
- `headerValues (String - String[])`
- `cookie (String - String)`

Java tárolók

tömb, Vector, List, Set, Map

```
<%  
    String[] allatok={"kutya", "macska", "elefant"};  
    request.setAttribute("allatok", allatok);  
%>
```

```
<p>Első: ${allatok[0]}</p>  
<p>Első: ${requestScope.allatok[0]}</p>
```

```
<p>Második: ${allatok['1']}</p>  
<p>Második: ${requestScope.allatok['1']}</p>
```

```
<p>Harmadik: ${allatok["2"]}</p>  
<p>Harmadik: ${requestScope.allatok["2"]}</p>
```

Kérés

```
index.jsp?productid=12&productid=23
```

```
`${paramValues.productid[0]}`
```

```
`${paramValues.productid[1]}`
```

```
`${paramValues['productid'][0]}`
```

```
`${paramValues['productid'][1]}`
```

```
<jsp:useBean id="person" class="model.Person"/>  
<jsp:setProperty name="person"  
    property="firstName" value="Margit"/>  
<jsp:setProperty name="person"  
    property="lastName" value="Antal"/>
```

```
<p>First name: ${person.firstName} </p>  
<p>Last name: ${person.lastName} </p>
```

```
${person[firstName]}  
${person['firstName']}  
${person["firstName"]}
```

Aritmetikai operátorok

Operátor neve	Operátor
összeadás	+
kivonás	-
szorzás	*
osztás	/
osztási hányados	%

Relációs és egyenlőségi operátorok

Operátor neve	Operátor
egyenlő	== és eq
különböző	!= és ne
kisebb	< és lt
nagyobb	> és gt
kisebb vagy egyenlő	<= és le
nagyobb vagy egyenlő	>= és ge

Relációs és egyenlőségi operátorok

Operátor neve	Operátor
és	&& és and
vagy	és or
nem	! és not


```
<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <el-ignored>>true</el-ignored>
  </jsp-property-group>

  <jsp-property-group>
    <url-pattern>/scriptless/*</url-pattern>
    <scripting-invalid>>true</scripting-invalid>
  </jsp-property-group>
</jsp-config>
```