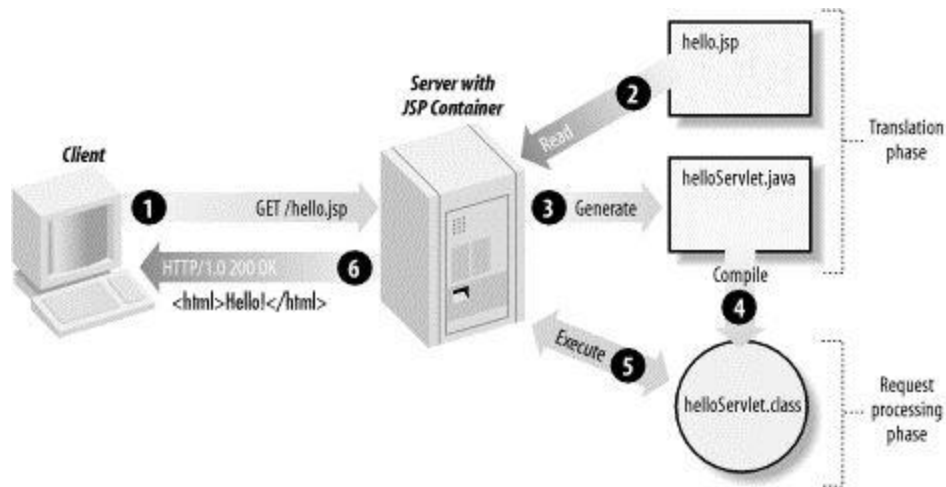
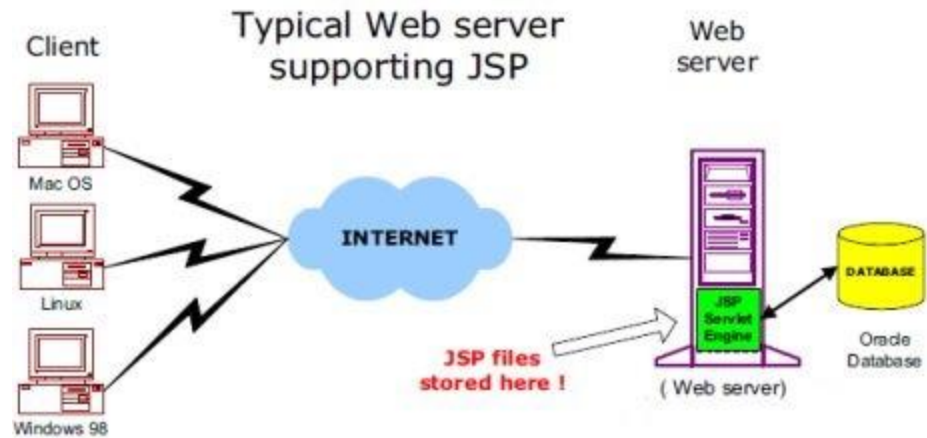


DEZVOLTAREA APLICATIILOR WEB

LAB 4

Lect. Univ. Dr. Mihai Stancu

➤ JSP Architecture



➤ Compilation

- Parsing the JSP.
- Turning the JSP into a servlet.
- Compiling the servlet

➤ Initialization

- `public void jspInit()`

➤ Execution

- `void _jspService(HttpServletRequest request, HttpServletResponse response)`

➤ Cleanup

- `public void jspDestroy()`

➤ The Scriptlet

- `<% code fragment %>`
- `<jsp:scriptlet> code fragment </jsp:scriptlet>`

```
<html>
<head><title>Hello World</title></head>
<body>
Hello World!<br/>
<%
out.println("Your IP address is " +
request.getRemoteAddr());
%>
</body>
</html>
```

➤ JSP Declarations

- `<%! declaration; [declaration;]+ ... %>`
- `<%! int i = 0; %>`
- `<%! int a, b, c; %>`
- `<%! Circle a = new Circle(2.0); %>`

➤ JSP Expression

- `<%= expression %>`
- `<jsp:expression> expression </jsp:expression>`

```
<html>
<head><title>A Comment Test</title></head>
<body>
<p>
  Today's date: <%= (new
java.util.Date()).toLocaleString()%>
</p>
</body>
</html>
```

➤ JSP Comments

- `<%-- This is JSP comment --%>`
- `<!-- This is HTML comment -->`
- `<\% , %\>` - static literals
- `\', \"` - static literals

➤ JSP Directives

➤ `<%@ directive attribute="value" %>`

Directive	Description
<code><%@ page ... %></code>	Defines page-dependent attributes, such as scripting language, error page, and buffering requirements.
<code><%@ include ... %></code>	Includes a file during the translation phase.
<code><%@ taglib ... %></code>	Declares a tag library, containing custom actions, used in the page

➤ JSP Actions

➤ `<jsp:action_name attribute="value" />`

➤ JSP Actions

Syntax	Purpose
jsp:include	Includes a file at the time the page is requested
jsp:useBean	Finds or instantiates a JavaBean
jsp:setProperty	Sets the property of a JavaBean
jsp:getProperty	Inserts the property of a JavaBean into the output
jsp:forward	Forwards the requester to a new page
jsp:plugin	Generates browser-specific code that makes an OBJECT or EMBED tag for the Java plugin
jsp:element	Defines XML elements dynamically.
jsp:attribute	Defines dynamically defined XML element's attribute.
jsp:body	Defines dynamically defined XML element's body.
jsp:text	Use to write template text in JSP pages and documents.

➤ JSP Implicit Objects

Objects	Description
request	This is the HttpServletRequest object associated with the request.
response	This is the HttpServletResponse object associated with the response to the client.
out	This is the PrintWriter object used to send output to the client.
session	This is the HttpSession object associated with the request.
application	This is the ServletContext object associated with application context.
config	This is the ServletConfig object associated with the page.
pageContext	This encapsulates use of server-specific features like higher performance JspWriters .
page	This is simply a synonym for this , and is used to call the methods defined by the translated servlet class.
Exception	The Exception object allows the exception data to be accessed by designated JSP.

➤ Control-Flow Statements

➤ Java blocks: decision making statements, loops, operators etc

```
<%! int day = 3; %>
<html>
<head><title>IF...ELSE Example</title></head>
<body>
<% if (day == 1 | day == 7) { %>
    <p> Today is weekend</p>
<% } else { %>
    <p> Today is not weekend</p>
<% } %>
</body>
</html>
```

```
<%! int fontSize; %>
<html>
<head><title>WHILE LOOP Example</title></head>
<body>
<%while ( fontSize <= 3){ %>
    <font color="green" size="<%= fontSize %>">
        JSP Tutorial
    </font><br />
<%fontSize++;%>
<%}%>
</body>
</html>
```

➤ JSP literals

- Boolean: true and false
- Integer: as in Java
- Floating point: as in Java
- String: with single and double quotes; " is escaped as \", ' is escaped as \', and \ is escaped as \\.
- Null: null

➤ The page Directive

- `<%@ page attribute="value" %>`
- `<jsp:directive.page attribute="value" />`

➤ The include Directive

- `<%@ include file="relative url" >`
- `<jsp:directive.include file="relative url" />`

➤ The taglib Directive

- `<%@ taglib uri="uri" prefix="prefixOfTag" >`
- `<jsp:directive.taglib uri="uri"
prefix="prefixOfTag" />`

➤ Common attributes: id, scope (page, request, session, application)

➤ The `<jsp:include>` Action

➤ `<jsp:include page="relative URL" flush="true" />`

Attribute	Description
page	The relative URL of the page to be included.
flush	The boolean attribute determines whether the included resource has its buffer flushed before it is included.

➤ `date.jsp`

```
<p>
  Today's date: <%= (new java.util.Date()).toLocaleString() %>
</p>
```

➤ `main.jsp`

```
<html>
<head>
<title>The include Action Example</title>
</head>
<body>
<center>
<h2>The include action Example</h2>
<jsp:include page="date.jsp" flush="true" />
</center>
</body>
</html>
```

➤ The <jsp:useBean> Action

➤ `<jsp:useBean id="name" class="package.class" />`

Attribute	Description
class	Designates the full package name of the bean.
type	Specifies the type of the variable that will refer to the object.
beanName	Gives the name of the bean as specified by the instantiate () method of the java.beans.Beans class.

➤ The <jsp:setProperty> Action

```
<jsp:useBean id="myName" ... />
...
<jsp:setProperty name="myName" property="someProperty" .../>
```

```
<jsp:useBean id="myName" ... >
...
  <jsp:setProperty name="myName" property="someProperty" .../>
</jsp:useBean>
```

Attribute	Description
name	Designates the bean whose property will be set. The Bean must have been previously defined.
property	Indicates the property you want to set. A value of "*" means that all request parameters whose names match bean property names will be passed to the appropriate setter methods.
value	The value that is to be assigned to the given property. The the parameter's value is null, or the parameter does not exist, the setProperty action is ignored.
param	The param attribute is the name of the request parameter whose value the property is to receive. You can't use both value and param, but it is permissible to use neither.

➤ The <jsp:getProperty> Action

```
<jsp:useBean id="myName" ... />
...
<jsp:getProperty name="myName" property="someProperty" .../>
```

Attribute	Description
name	The name of the Bean that has a property to be retrieved. The Bean must have been previously defined.
property	The property attribute is the name of the Bean property to be retrieved.

```
/* File: TestBean.java */
package action;

public class TestBean {
    private String message = "No message specified";

    public String
        return(mess
    }
    public void se
        this.messag
    }
}
```

```
<html>
<head><title>Using JavaBeans in JSP</title></head>
<body>
    <center>
        <h2>Using JavaBeans in JSP</h2>

        <jsp:useBean id="test" class="action.TestBean" />
        <jsp:setProperty name="test" property="message" value="Hello JSP..." />

        <p>Got message....</p>
        <jsp:getProperty name="test" property="message" />

    </center>
</body>
</html>
```

➤ The <jsp:forward> Action

```
<jsp:forward page="Relative URL" />
```

Attribute	Description
page	Should consist of a relative URL of another resource such as a static page, another JSP page, or a Java Servlet.

➤ date.jsp

```
<p>  
Today's date: <%= (new java.util.Date()).toLocaleString() %>  
</p>
```

➤ main.jsp

```
<html>  
<head>  
<title>The include Action Example</title>  
</head>  
<body>  
  <center>  
    <h2>The include action Example</h2>  
    <jsp:forward page="date.jsp" />  
  </center>  
</body>  
</html>
```


➤ The <jsp:element>, <jsp:attribute>, <jsp:body> Actions

➤ define XML elements dynamically (at request time)

➤ xml.jsp

```
<%@page language="java" contentType="text/html"%>
<html xmlns="http://www.w3c.org/1999/xhtml" xmlns:jsp="http://java.sun.com/JSP/Page">

<head><title>Generate XML Element</title></head>
<body>
  <jsp:element name="xmlElement">
    <jsp:attribute name="xmlElementAttr">
      Value for the attribute
    </jsp:attribute>
    <jsp:body>
      Body for XML element
    </jsp:body>
  </jsp:element>
</body>
</html>
```

➤ output

```
<html xmlns="http://www.w3c.org/1999/xhtml" xmlns:jsp="http://java.sun.com/JSP/Page">

<head><title>Generate XML Element</title></head>
<body>
<xmlElement xmlElementAttr="Value for the attribute">
  Body for XML element
</xmlElement>
</body>
</html>
```

➤ The <jsp:text> Action

- contine decat text si expresii EL (\${whatever > 0})

```
<jsp:text>Template data</jsp:text>
```

- in XML < > se substituie cu lt gt

```
<jsp:text><![CDATA[<br>]]></jsp:text>
```

```
<jsp:text><![CDATA[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"DTD/xhtml11-strict.dtd">]]>
</jsp:text>
<head><title>jsp:text action</title></head>
<body>

<books><book><jsp:text>
  Welcome to JSP Programming
</jsp:text></book></books>

</body>
</html>
```

Attribute	Description
request	This is the HttpServletRequest object associated with the request.
response	This is the HttpServletResponse object associated with the response to the client.
out	This is the PrintWriter object used to send output to the client.
session	This is the HttpSession object associated with the request.
application	This is the ServletContext object associated with application context.
config	This is the ServletConfig object associated with the page.
pageContext	This encapsulates use of server-specific features like higher performance JspWriters .
page	This is simply a synonym for this , and is used to call the methods defined by the translated servlet class.
Exception	The Exception object allows the exception data to be accessed by designated JSP.

➤ The config Object

```
config.getServletName();
```

➤ The pageContext Object

```
pageContext.removeAttribute("attrName", PAGE_SCOPE);
```

- PAGE_SCOPE, REQUEST_SCOPE, SESSION_SCOPE, APPLICATION_SCOPE

JSP – CLIENT REQUEST

```
<%@ page import="java.io.*,java.util.*" %>
<html>
<head>
<title>HTTP Header Request Example</title>
</head>
<body>
<center>
<h2>HTTP Header Request Example</h2>
<table width="100%" border="1" align="center">
<tr bgcolor="#949494">
<th>Header Name</th><th>Header Value(s)</th>
</tr>
<%
    Enumeration headerNames = request.getHeaderNames();
    while(headerNames.hasMoreElements()) {
        String paramName = (String)headerNames.nextElement();
        out.print("<tr><td>" + paramName + "</td>\n");
        String paramValue = request.getHeader(paramName);
        out.println("<td> " + paramValue + "</td></tr>\n");
    }
%>
</table>
</center>
</body>
</html>
```

JSP – SERVER RESPONSE

```
<%@ page import="java.io.*,java.util.*" %>
<html>
<head>
<title>Auto Refresh Header Example</title>
</head>
<body>
<center>
<h2>Auto Refresh Header Example</h2>
<%
    // Set refresh, autoloading time as 5 seconds
    response.setIntHeader("Refresh", 5);
    // Get current time
    Calendar calendar = new GregorianCalendar();
    String am_pm;
    int hour = calendar.get(Calendar.HOUR);
    int minute = calendar.get(Calendar.MINUTE);
    int second = calendar.get(Calendar.SECOND);
    if(calendar.get(Calendar.AM_PM) == 0)
        am_pm = "AM";
    else
        am_pm = "PM";
    String CT = hour+":"+minute+":"+second+" "+am_pm;
    out.println("Current Time is: " + CT + "\n");
%>
</center>
</body>
</html>
```

JSP – HTTP STATUS CODES

```
<html>
<head>
<title>Setting HTTP Status Code</title>
</head>
<body>
<%
    // Set error code and reason.
    response.sendError(407, "Need authentication!!!" );
%>
</body>
</html>
```

➤ GET

➤ `http://localhost:8080/main.jsp?first_name=ZARA&last_name=ALI`

```
<html>
<head>
  <title>Using GET Method to Read Form Data</title>
</head>
<body>
  <center>
    <h1>Using GET Method to Read Form Data</h1>
    <ul>
      <li><p><b>First Name:</b>
        <%= request.getParameter("first_name")%>
      </p></li>
      <li><p><b>Last Name:</b>
        <%= request.getParameter("last_name")%>
      </p></li>
    </ul>
  </center>
</body>
</html>
```

➤ GET

➤ Using Form

```
<html>
<body>
  <form action="main.jsp" method="GET">
    First Name: <input type="text" name="first_name">
    <br />
    Last Name: <input type="text" name="last_name" />
    <input type="submit" value="Submit" />
  </form>
</body>
</html>
```

➤ POST

```
<html>
<body>
  <form action="main.jsp" method="POST">
    First Name: <input type="text" name="first_name">
    <br />
    Last Name: <input type="text" name="last_name" />
    <input type="submit" value="Submit" />
  </form>
</body>
</html>
```


➤ Passing Checkbox Data

```
<html>
<body>
  <form action="main.jsp" method="POST" target="_blank">
    <input type="checkbox" name="maths" checked="checked" /> Maths
    <input type="checkbox" name="physics" /> Physics
    <input type="checkbox" name="chemistry" checked="checked" /> Chemistry
    <input type="submit" value="Select Subject" />
  </form>
</body>
</html>
```

```
<html>
<head>
<title>Reading Checkbox Data</title>
</head>
<body>
  <center>
    <h1>Reading Checkbox Data</h1>
    <ul>
      <li><p><b>Maths Flag:</b> <%= request.getParameter("maths")%>
        </p></li>
      <li><p><b>Physics Flag:</b> <%= request.getParameter("physics")%>
        </p></li>
      <li><p><b>Chemistry Flag:</b> <%= request.getParameter("chemistry")%>
        </p></li>
    </ul>
  </center>
</body>
</html>
```

➤ Reading All Form Parameters

```
<%@ page import="java.io.*,java.util.*" %>
<html>
<head>
  <title>HTTP Header Request Example</title>
</head>
<body>
  <center>
    <h2>HTTP Header Request Example</h2>
    <table width="100%" border="1" align="center">
      <tr bgcolor="#949494">
        <th>Param Name</th><th>Param Value(s)</th>
      </tr>
<%
  Enumeration paramNames = request.getParameterNames();

  while(paramNames.hasMoreElements()) {
    String paramName = (String)paramNames.nextElement();
    out.print("<tr><td>" + paramName + "</td>\n");
    String paramValue = request.getHeader(paramName);
    out.println("<td> " + paramValue + "</td></tr>\n");
  }
%>
    </table>
  </center>
</body>
</html>
```

HOMWORK

- Homework
 - jsp template app