Instituto Politécnico Nacional

Escuela Superior de Cómputo

THEMATIC UNIT: III

Java Server Pages

M. en C. José Asunción Enríquez Zárate

asuncionez@gmail.com

Java Server Pages

UNIT OF COMPETENCE

The student builds Web applications based on Java Server Pages specification.

Contents

Contents

JSP, Servlet and JEE Architecture

Java Server Pages

Extensible web technology that uses custom elements, scripting languages, and java server objects to return dynamic content to the client, typically an html or xml.

Servlets

A Java Servlet is a java program that allows to generate dynamic content and interact with the client.

Java Server Pages

Diferent Content types

- Static Content
  - html pages.
  - The same GUI for any page.
- Dynamic Content.
  - The content is dynamically generated based on conditions.
  - The conditions could be:
    - User Identity.
    - Time of the Day.
    - Values user entered.

Contents

Contents

Directives

- Appear at the top of the page.
- Contain special processing instructions for the web container.
- examples:
  - <%@page import="java.util.List"%>
  - <%@taglib prefix="c"
    uri="http://java.sun.com/jsp/jstl/core"%>
  - <%@page import="com.ipn.mx.dao.EventoDAO"%>

Declarations

- Variables and methods can be declared using the
  - <%! and %> tags.
- Anything declared within these tags will be visible to the rest of the page.
- examples:

```
1   <%!
      int balance = 1000;
3     public int getBalance(){
        return balance;
5     }
    %>
7
```

- the account balance is: <%= getBalance() %>

Declarations

- Variables and methods can be declared using:

```
<jsp:declaration>
  double  interest  = 0;
  public double calculateInterest(){
    interest = balance * 0.20 * 12;
    return interest;
  }\justifying
</jsp:declaration>
```

- The interest in a year is: <%= calculateInterest() %>

Expressions

- Is an instruction to the web container to execute the code within the expression and replace it with the results in the outputted content.
- Expressions must be declared betwen <%= and %> tags.
- The interest in a year is: <%= calculateInterest() %>

Scriptlets

- Its possible use scriptlets anywhere in a page.
- They're fragments of Java code.
- Anything that can be done inside of a servlet can go inside of a scriptlet.
- Scriptlets must be declared betwen <% and %> tags.

```
String userName = request.getParameter("username");
String userPass = request.getParameter("userPassword");
if (userName.equals("admin") && userPass.equals("admin")){
  out.println("welcome");
  HttpSession session = request.getSession();
  session.setAttribute("theSession",userName);
  response.sendRedirect("mainPage.jsp");
}else{
  response.sendRedirect("index.jsp");
}
```

Handling Errors

- JSP has an elegant solution for error handling.
  - With erorPage directive
    - <%@page isErrorPage="true" %>
  - use
    - <%@page errorPage="error.jsp?from=login.jsp" %>
  - The from parameter is using to identificate which page produces the exception

error.jsp

```
<%@page isErrorPage="true" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!doctype html>
<html>
  <head>
    <title> Error Page </title>
    <link href="css/estilos.css" type="text/css" rel="stylesheet"/>
  </head>
  <body>
    <br/>
    <%
      String from = (String)request.getParameter("from");
    %>
    <p class="tituloError">
    An error occurred on page <b><i class="error"><%= from %>
    </i></b>.
    <br/><br/>
    The exception was:
    <br/>
      <b><i class="error"><%= exception %></i>
    </b>
  </body>
</html>

```

Using a handling error page

- When you've created an error page, you need to code your pages to forward all uncaught exceptions to it.
- You do this by adding the JSP page directive errorPage.
- Each page you want to forward errors from needs this directive included.
  - Using the page error.jsp
    - ***eror.jsp***, is expecting a parameter named ***from*** to determine which page called the error page.

productList.jsp

```
1   <%@page import="com.ipn.modelo.dto.ArticuloDTO"%>
    <%@page import="java.util.List"%>
3   <%@page import="com.ipn.modelo.delegate.EzjaMVCDelegate"%>
    <%@page errorPage="error.jsp?de=productList.jsp"  %>
5   <%@page contentType="text/html" pageEncoding="UTF-8"%>
    <!doctype html>
7   <html>
      <head>
9       <title>Product List</title>
        <link type="text/css" href="css/estilos.css" rel="stylesheet"/>
11    </head>
      <body>
13      <table class="tabla">
          <%
15          EzjaMVCDelegate del = new EzjaMVCDelegate();
            List lista = del.listarArticulos();
17          if (lista != null) {
              for (int i = 0; i < lista.size(); i++) {
19              ArticuloDTO listaArticulo = (ArticuloDTO) lista.get(i);
          %>
21    <tr>

23
```

productList.jsp cont..

```
            <tr>
2             <td> <%= listaArticulo.getClaveArticulo()%> </td>
              <td> <%= listaArticulo.getDescripcion()%> </td>
4             <td> <%= listaArticulo.getExistencias()%> </td>
              <td> <%= listaArticulo.getPrecio()%> </td>
6             <td>
                 <a href="eliminar.jsp?id=<%=listaArticulo.getClaveArticulo
      ()%>">
8                  Eliminar
                </a>
10            </td>
            </tr>
12        <%
                }
14            } else {
              out.println("<td>No hay Registros </td>");
16            }
        %>
18    </table>
  </body>
20  </html>
```

Contents

EL Scopes

- Data storage locations:
  - The request object
  - The session object
  - Servlet initialization parameters
  - Application initialization parameters Scopes
- Servlet context (application) also offers a read-write map
- Three scopes are request, session, application
- In a servlet's doXxx methods, local variables are an option
- In a JSP, a page scope exists roughly equivalent to local variables
- Use of page scope is dubious in an MVC design.

EL Scopes

Scope Name	Communication
page	Between local variables within a JSP only. Equivalent to local variables in a `doXxx` servlet method.
request	Between components cooperating in the execution of a single request from a single browser. Typically used to carry data from a controller servlet to the view JSP.
session	Between servlets and JSPs used during a single user session, across the different requests being made.
application	Between all components of a single application.

Figure: Scopes Sumary.

EL

- Basic form: ${expr}
- Escape EL processing with backslash:\${literal}
- EL scans for names in all four scopes
- Search order is: page, request, session, application
- Explicit reference overrides search order ${sessionScope.customer.firstName}

EL

Implicit Object	Description
pageContext	The PageContext objec
pageScope	A map containing page-scoped attributes and their values
requestScope	A map containing session-scoped attributes and their values
applicationScope	A map containing application-scoped attributes and their values
param	A map containing request parameters and single string values

Table: EL Implicit objects.

Scopes

Implicit Object	Description
paramValues	A map containing request parameters and their corresponding string arrays
header	A map containing header names and single string values
headerValues	A map containing header names and their corresponding string arrays
cookie	A map containing cookie names and their values
initParam	A map of the servlet's init parameters

Use implicit objects as: ${param.username}

The Dot Operator in EL

- Dot operator access either JavaBean property or a map element
- Legitimate keys must adhere to JavaBeans naming conventions
  - A map key including dot would fail

EL Arithmetic Operators

EL Arithmetic Operators

- Type conversion is performed where necessary and possible

Arithmetic Operation	Operator
Addition	+
Subtraction	–
Multiplication	*
Division	/ and div
Remainder	% and mod

EL Expression	Result
${3 div 4}	0.75
${1 + 2 * 4}	9
${(1 + 2) * 4}	12
${32 mod 10}	2

Figure: EL Arithmetic Operators.

Comparisons and Logical Operators

Comparisons and Logical Operators

- Comparison operators

Comparison	**Operator**
Equals	== and `eq`
Not equals	!= and `ne`
Less than	< and `lt`
Greater than	> and `gt`
Less than or equal	<= and `le`

Figure: Comparison operators.

Comparisons and Logical Operators

Comparisons and Logical Operators

- Logical operators

Operation	**Operator**
and	`&&` and `and`
or	`||` and `or`
not	`!` and `not`
Test if an array or list is empty	`empty`

Figure: Comparison operators.

Java Standard Tag Library

Presentation Programming

- Presentation sometimes needs behavior
  - Iteration over a list or table.
  - Selection of conditional elements
- The Java Standard Tag Library or JSTL addresses these needs.

Java Standard Tag Library

JSTL if Tag

```
<c:if test="expression" var="varName"
 [scope="{page|request|session|application}"] >
 body evaluated if expression evaluates to true
</c:if>
```

Figure: If JSTL.

Java Standard Tag Library

```
<c:forEach items="collection" [var="varName"]
  [varStatus="varStatusName"]
  [begin="begin"] [end="end"] [step="step"]>
  body content
</c:forEach>
```

JSTL forEach Tag

- Collection may be
  - Collection.
  - Iterator.
  - Map.
  - Enumeration.
  - Array.
  - Comma separated string.

Java Standard Tag Library - JSTL forEach Tag

Attribute	Use
var	Specifies the name of the nested-visibility variable, which will contain the current element in the iteration.
varStatus	Specifies the name of the nested-visibility variable, which will contain the status of the current cycle in the iteration. varStatus is itself a structured type and has elements index and count, that track the progress through the loop
begin	Specifies the index of the first item in the iteration.
end	Specifies the last item in the iteration.
step	Skips over elements in the iteration.

Contents

Referencias

Oracle.
Web Component Development With Servlet and JSP Technologies
Oracle.

Edgar Martinez, Tom McGinn, Eduardo Moranchel, Anjana Shenoy, Michael Williams.
Java EE 7: Back-end Server Application Development
Oracle, 2016.

Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito.
ava Platform, Enterprise Edition The Java EE Tutorial, Release 7
Oracle, 2014.