- Java Server Pages -

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS

- Directivas JSP.
- Elementos de Script.
- Acciones.

DEFINICIÓN

> Especificación de SUN para la elaboración de **páginas dinámicas** usando el lenguaje de programación **JAVA**

OBJETIVO

> Separar la interfaz (presentación) de la implementación (lógica de aplicación)

OBJETIVO

Usando Servlets

- Leer datos formulario.
- HTTP (Cabeceras, estados...).
- Cookies y Sesiones.
- Compartir datos con Servlets.
- Recordar datos entre peticiones.

OBJETIVO

Usando Servlets

- Muchas Instrucciones println() para generar salida HTML.
- Mantenimiento HTML.

FUNCIONAMIENTO

- La página jsp se convierte en un Servlet al hacer la primera petición.
- El servlet generado procesa las peticiones.
- Si modificamos el JSP se regenera automáticamente el Servlet con la nueva versión para la siguiente petición.

EJEMPLO HOLA MUNDO

```jsp
<%@ page contentType="text/html;charset=UTF-8" language="java" import="java.util.Date" %>
<html>
<head>
    <title>Hola mundo en JSP</title>
</head>
<body>
    <h1>Hola mundo en JSP</h1>
    <p>La fecha de hoy es: <%= new Date() %>  </p>
</body>
</html>
```

ETAPAS DE CONVERSIÓN

1. Traducción a **.java**
2. Compilación (**.class**)
3. Carga de la clase.
4. Se crea la instancia y comienza el ciclo de vida del servlet (**init**, **service**, **destroy**)

ELEMENTOS DE PÁGINA JSP

- Objetos implícitos (request, response, session, out, config, application...)
- Directivas.
- Elementos de Script.

Objetos implícitos

Objeto	Contexto	Descripción	Ámbito
request	HttpServletRequest	Invocación del servicio	Petición
response	HttpServletResponse	Respuesta a la Petición	Página
pageContext	jsp.PageContext	Características de la página Dependientes de la implementación, espacios de nombres y otras facilidades	Página
session	http.HttpSession	Conserva el estado de la session	Sesión
application	ServletContext	Contexto del servlet de configuración	Aplicación
out	jsp.JspWriter	Flujo de salida	Página
config	ServletConfig	Configuración del servlet del JSP	Página
page	Object	Página que procesa la petición en curso	Página

DIRECTIVAS JSP

- **Page** (Configuración de la página).
- **Include** (Añadir otros ficheros **HTML, JSP**...)
- **Taglib** para añadir librerías de etiquetas generadas por el propio usuario (se verá más adelante....)

DIRECTIVAS JSP

Page <%@ page ATRIBUTOS %>

- import
- contentType
- encoding
- errorPage

```
<%@ page
contentType="text/html;charset=UTF-8"
language="java"
import="java.util.Date"
%>
```

DIRECTIVAS JSP

include <%@ include file=”.....” %>

```
<html>
<head>
    <title>Prueba de include</title>
</head>
<body>
    <h1>Prueba de include</h1>
    <%@ include file="fichero.html"  %>
    <%@ include file="fichero.jsp"  %>
</body>
</html>
```

ELEMENTOS DE SCRIPT

- **Declaraciones:** Para declarar variables y funciones con ámbito en el servlet <%! … %>
- Código de Java (**Scriptlets**). <% … %>
- **Expresiones:** Para Mostra por pantalla el resultado de una expresión a evaluar
  <%= …%>

ELEMENTOS DE SCRIPT

- Los elementos de Scripting llaman al código del Servlet **DIRECTAMENTE**
- Los elementos de Scripting llaman al código del Servlet **INDIRECTAMENTE** mediante clases de uso general (utilities)
- Uso de **Beans**
- Etiquetas personalizadas
- **Servlet/JSP** combo

Aplicaciones simples

Aplicaciones complejas

ACCIONES

> **Etiquetas**, con formato **XML** que afectan al **comportamiento** en ejecución del **JSP**. Al traducirlos a **Servlet** se reemplazan por código **Java** (como si fuera una macro).

ACCIONES

Existen unas estándar pero el usuario puede definir las propias.

<prefijo:sufijo atributos />

Ejemplo:

```
<jsp:include page="fichero.jsp" />
```

Acciones estándar:

- <jsp:include>
- <jsp:param>
- <jsp:forward>
- <jsp:useBean>
- <jsp:setProperty>
- <jsp:getProperty>

ACCIONES

<jsp:param>

Para pasar parámetros de una página a otra (como si fuera parte de un formulario)

```
<jsp:include page="fichero.jsp">
    <jsp:param name="nombre" value="valor"/>
</jsp:include>
```

Dentro de fichero.jsp

```
<% request.getParameter("nombre"); %>
```

ACCIONES

<jsp:forward>

- La petición (**request**) es Redireccionada a otra página, otro **servlet** o recurso estático.
- Útil si quiero separar la aplicación en diferentes vistas dependiendo de la petición.
- No se vuelve a la original.

ACCIONES

<jsp:forward>

```
<jsp:forward page="URL">
    <jsp:param name="nombre" value="valor"/>
        <%-- no obligatorios--%>
</jsp:forward>
```

ACCIONES

<jsp:forward>

```
<body>
    <form method="post" action="fichero.jsp">
        Usuario: <input type="text" name="username"><br>
        Clave: <input type="text" name="password"><br>
                <input type="submit" name="login" value="Login">
    </form>
</body>
```

ACCIONES

<jsp:forward>

```
<%
    if ((request.getParameter("userName").equals("loQueSea"))
    && (request.getParameter("password").equals("quienSabe"))) {
%>
    <jsp:forward page="saludoforward.jsp"/>
<% } else { %>
    <%@ include file="forward.html"%>
<% } %>
```

ACCIONES

<jsp:bean>

- Los beans son objetos **Java** serializables y con un constructor sin argumentos.
- Me permite separar la lógica (el **Bean**) de la presentación (**servlet - vista**).
- Se crea la clase aparte y se instancia y usa en el **JSP**.
- Los atributos se acceden mediante **getNombre/setNombre**.

ACCIONES

<jsp:useBean>

```
<jsp:useBean id="nombre" scope="ambito" class="clase"/>

<jsp:useBean id="cart" scope="session" class="pkg.cart"/>
```

<jsp:useBean> - Ámbitos

- page.
- request.
- session.
- application.

ACCIONES

<jsp:setProperty>

- Junto con **useBean** para asignar valor a las propiedades del **Bean**.
- Se ejecuta en el **_jspService(..)** del servlet generado y se invoca al **setXXX** de la propiedad.

ACCIONES

<jsp:setProperty>

```
<jsp:setProperty name="id-bean"
property="propiedad"/>
```

- property=”*” Todas los parámetros de request
- property=”Nombre” Solo ese parámetro de request
- property=”Nombre” param=”NombreParam”
- property=”Nombre” value=”...”

ACCIONES

<jsp:getProperty>

- **Para obtener el valor de las propiedades del Bean**
- **Se ejecuta en el _jspService(..) del servlet generado, convierte a String y lo imprime por la salida del cliente (out).**

```
<jsp:getProperty name="id-bean" property="propiedad"/>
```

END

jgardur081@g.educaand.es