

Java Server Pages (JSP)

Laboratorio de Aplicaciones Telemáticas (Curso 2009/2010)

Jesús Arias Fisteus

jaf@it.uc3m.es



Introducción

- Un servlet no resulta adecuado para presentación (escribir directamente la salida HTML) porque el código HTML está entremezclado dentro del código Java:
 - Incómodo cambiar el código HTML.
 - No facilita la división de tareas entre diseñadores de HTML y programadores.
- Alternativa: escribir directamente el código HTML con poco código Java incrustado.

Introducción

- Una página JSP es una página (X)HTML que incorpora ciertos elementos dinámicos: etiquetas especiales y pequeños fragmentos de código.
 - El código HTML aparece a la salida sin modificaciones.
 - Los elementos dinámicos se evalúan o ejecutan en el servidor en el momento de construcción de la respuesta.

JSP como servlet

- Aunque no es estrictamente obligatorio, una página JSP se suele transformar en el código fuente de un servlet, que después se compila y ejecuta.

JSP ejemplo

```
<%@ page language='java' contentType='text/html; charset=iso-8859-1' %>
<%@ page import='java.util.Date' %>
<html>
  <head>
    <title>Hola Mundo</title>
  </head>
  <body>
    <p>Hola, esto es una página JSP.</p>
    <p>La hora del servidor es <%= new Date() %></p>
  </body>
</html>
```

JSP ejemplo transformado en un servlet

```
(...)  
out.write("<html>\n ");  
out.write("  <head>\n ");  
out.write("    <title>Hola Mundo</title>\n ");  
out.write("  </head>\n ");  
out.write("  <body>\n    <p>Hola, esto es una página JSP.</p>\n ");  
out.write("    <p>La hora del servidor es ");  
out.print( new Date() );  
out.write("</p>\n ");  
out.write("  </body>\n");  
out.write("</html>\n");  
(..)
```

JSP ejemplo: documento recibido por el cliente

```
<html>
  <head>
    <title>Hola Mundo</title>
  </head>
  <body>
    <p>Hola, esto es una página JSP.</p>
    <p>La hora del servidor es Wed Nov 06 13:25:34 CET 2002</p>
  </body>
</html>
```

Comentarios en JSP

- Los comentarios JSP no se envían al cliente, al contrario que los comentarios HTML.

```
<%!-- Esto es un comentario JSP --%>
```

Directivas JSP: *page*

- Todas las páginas JSP deberían incluirla.
Atributos habituales:
 - `language`: lenguaje de programación (java por defecto).
 - `contentType`: tipo de contenido de la página (`text/html` por defecto).
 - `isErrorPage`: indica si es una página de error (`false` por defecto).
 - `errorPage`: página a la que dirigirse si ocurre una excepción procesando esta página.

```
<%@ page language='java' contentType='text/html'
      isErrorPage='false' errorPage='error.jsp' %>
```

Otras directivas JSP

- `include`: permite incluir directamente el código de otro fichero en el punto en que aparezca la directiva.
- `import`: permite importar clases Java utilizadas en la página JSP.

```
<%@ include file='footer.html' %>  
<%@ page import='java.util.*' %>
```

Scriptlets

- Permiten incrustar código escrito en otro lenguaje de programación (normalmente Java):
 - `<%= expresión %>`: evalúa la expresión y muestra el resultado en la página.
 - `<% sentencias %>`: ejecuta las sentencias, sin mostrar nada en la página.
 - `<%! declaraciones %>`: declaraciones de variables.

```
<%!-- los siguientes scriptlets son equivalentes --%>  
<%= user.getName() %>  
<% out.println(user.getName()); %>
```

Scriptlets (aviso)

- Con `<%! declaraciones %>` se declaran atributos y métodos de instancia:
 - Todas las peticiones a la página JSP comparten la misma instancia del servlet asociado, por lo que también comparten la misma copia de los atributos de instancia.
 - Por tanto, **no se debe** declarar variables locales de la página con esta directiva, sino con:

```
<% String text = new String(); %>
```

Scriptlets: ejemplo

```
<table>
  <tr><th>Product</th><th>Price</th></tr>
  <%
    for (int i=0; i<catalog.length; i++ ) {
      ProductInfoBean product = catalog[i];
  %>
  <tr>
    <td>
      <a href='<%= response.encodeURL("view?id="
        + product.getId()) %>'>
        <%= product.getShortName() %>
      </a>
    </td>
    <td><%= product.getPrice() %> Euros</td>
  </tr>
  <% } %>
</table>
```

Variables implícitas

- JSP proporciona automáticamente una serie de variables adecuadamente declaradas e inicializadas, utilizables por el programador:

```
javax.servlet.ServletContext application
javax.servlet.ServletConfig config
javax.servlet.jsp.JspWriter out
javax.servlet.jsp.PageContext pageContext
javax.servlet.http.HttpServletRequest request
javax.servlet.http.HttpServletResponse response
javax.servlet.http.HttpSession session
```

```
<%!-- Sólo en páginas marcadas con isErrorPage: --%>
java.lang.Throwable exception
```

Acciones: `jsp:include`

- La acción `jsp:include` invoca al servlet o JSP dado e incluye en resultado de su ejecución en el punto actual del documento JSP desde el cual se incluya.
- El control retorna finalmente a la página inicial.
- Opcionalmente, se pueden pasar parámetros.

```
<jsp:include page='header.jsp'>  
  <jsp:param name='title' value='Welcome' />  
</jsp:include>
```

Acciones: `jsp:forward`

- La acción `jsp:forward` invoca al servlet o JSP dado e incluye en resultado de su ejecución en el punto actual del documento JSP desde el cual se incluya, sin que retorne el control a la página inicial.
- Opcionalmente, se pueden pasar parámetros.

```
<jsp:forward page='list.jsp'>  
  <jsp:param name='order' value='date' />  
</jsp:forward>
```

Java Beans

- Un Java Bean es una clase Java que cumple el siguiente convenio:
 - Contiene *propiedades* (normalmente atributos de instancia privados).
 - El acceso a las propiedades se realiza mediante métodos de acceso *get*, *set* e *is*.
 - Tiene siempre un constructor sin argumentos (aunque podría tener más constructores).

Java Beans: ejemplo

```
public class UserInfoBean implements java.io.Serializable
{
    private String firstName;
    private boolean registered;

    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public boolean isRegistered() {
        return registered;
    }
    public void setRegistered(boolean registered) {
        this.registered = registered;
    }
}
```

Java Beans en JSP

- JSP proporciona instrucciones especiales para trabajar más cómodamente con Java Beans.
- La acción `jsp:useBean`:
 - Si el bean aún no existe en el contexto:
 - Declara, crea e inicializa el bean.
 - Crea una referencia al bean con el nombre dado por `id`.
 - Si el bean ya existe en el contexto:
 - Obtiene una referencia al mismo con el nombre dado por `id`.

```
<jsp:useBean id='user' class='foo.UserInfoBean' scope='session'>  
  <jsp:setProperty name='user' property='name' value='Pepe' />  
</jsp:useBean>
```

Java Beans en JSP: contextos (I)

- Un bean en JSP se puede almacenar en uno de los siguientes contextos:
 - `page`: asociado a la página JSP y a una petición HTTP concreta, desaparece al finalizar el procesamiento de la página.
 - `request`: asociado a una petición HTTP concreta, compartida entre todos los JSPs y servlets que atiendan dicha petición, desaparece al finalizar el procesamiento de la petición.

Java Beans en JSP: contextos (II)

■ . . .

- `session`: asociado a la sesión, compartida por todos los JSPs y servlets para todas las peticiones de una misma sesión, desaparece al finalizar la sesión.
- `application`: asociado al contexto de la aplicación Web, compartido por todos los servlets y JSPs de la aplicación en todas las peticiones.

La acción `jsp:getProperty`

- Se evalúa al valor de una propiedad de un bean.

```
<jsp:getProperty name='user' property='fullName' />
```

La acción `jsp:setProperty`

- Establece el valor de una propiedad de un bean.
- Convierte, si es necesario, el valor de la propiedad desde una cadena de texto al tipo de datos correspondiente.
- Proporciona un atajo para establecer valores de propiedades a partir de los parámetros de la petición, si ambos tienen el mismo nombre.

```
<jsp:setProperty name='user' property='firstName'
                value='<%= request.getParameter("firstName") %>' />
<%!-- si 'firstName' es un parámetro de la petición --%>
<jsp:setProperty name='user' property='firstName' />
<%!-- todos los parámetros de la petición cuyo nombre
        coincida con propiedades --%>
<jsp:setProperty name='user' property='*' />
```

Java Server Pages Tag Libraries

- Permiten declarar nuevas marcas JSP (*tags*) cuyo código Java se programa en clases de implementación de cada tag:
 - Permite reducir el código Java incrustado en páginas.
 - Útil para acciones utilizadas a menudo (p.e. acceso a bases de datos).
- No las veremos en esta asignatura.

Algunas referencias

`http://java.sun.com/products/jsp/`

`http://www.jsptut.com/`

`http://www.coreservlets.com/`

`http://www.programacion.com/java/tutorial/servlets_jsp/`