JSP(JAVA SERVER PAGES)

Introducción

En la actualidad, la mayoría de los sitios Web quieren mostrar contenido dinámico basado en el usuario y en la sesión. La mayor parte del contenido, tal como imágenes, texto y anuncios son más fáciles de crear en editores de HTML. Por lo tanto es necesario mezclar el contenido estático de archivos HTML con las directivas para acceder o generar contenido dinámico.

Las JSP cubren esta necesidad. Proveen soporte para el scripting(código fuente) desde el lado del servidor para generar páginas Web con contenido estático y dinámico combinado.

Pedidos y respuestas

Cuando un usuario ingresa un URL en un browser, es creado un pedido que va a un servidor. Dependiendo del pedido, la respuesta del servidor puede ser en forma de páginas estáticas HTML que están guardadas en el servidor o contenido dinámico que es conseguido de varias fuentes.

Hasta ahora, la forma más común de generar contenido dinámico era a través de CGI(Common Gateway Interface, Interfaz de compuerta común). Los programas CGI(escritos típicamente en C o Perl) interactuan con el usuario leyendo su entrada, en los formularios HTML, y devolviendo páginas HTML personalizadas. Pero la CGI tiene una desventaja: para cada pedido, el script CGI debe ser cargado, ejecutado y descargado, lo cual es ineficiente.

Respuestas estáticas y dinámicas

Otra forma de generar contenido dinámico es usar los servlets de Java. Los ***servlets*** son programas en Java que son cargados en un servidor de aplicaciones, tal como WebSphere, para extender sus capacidades. Se ejecuta en el contexto de una Java Virtual Machine. Hacen la misma cosa que los scripts CGI, pero residen en la memoria del servidor, por lo cual son más rápidos ante los pedidos de usuarios. Ya que los servlets son invocados por un pedido de usuario, representan la respuesta del sistema para una única localidad en un sitio Web. Una empresa puede tener varios servlets para requerimientos individuales como registro de clientes, compras, envíos, etc.

Las JSPs proveen una manera de combinar los mundos de HTML y la programación de servlets de Java. Las JSPs son archivos de texto que se parecen mucho a páginas HTML. El HTML es mejorado con nuevos tags que especifican la programación de un servidor para controlar la generación de contenido dinámico.

Si el usuario ha pedido información que esta en páginas estáticas que residen en el servidor http, la respuesta será una versión html de la página guardada. Para respuestas dinámicas, una llamada irá desde el servidor http a WebSphere, u otro servidor de aplicaciones, donde las JSPs y los servlets son manejados. El servidor de aplicaciones puede ser configurado para precargar los Java servlets para que puedan responder rápidamente hasta al primer pedido del usuario.

Los archivos jsp solo necesitan ser compilados y cargados una vez. En algún momento futuro, si una nueva versión aparece, el servidor de aplicación compilará la más nueva y cargará su versión del servlet.

Ventajas de las JSPs

Estas son razones por las que las JSPs simplifican la tarea de escribir páginas Web dinámicas

1. Java es conocido por su característica de "*escribir una vez, ejecutar en cualquier lugar*". Las JSPs son totalmente independientes de plataforma, ya sea en cuanto a las páginas Web dinámicas y a los componentes de servidor subyacentes. Se pueden escribir páginas dinámicas en cualquier plataforma, ejecutarlas en cualquier servidor Web y acceder a ellas desde cualquier navegador.

2. Las JSPs fomenta el *uso de componentes reusables de servidor llamados JavaBeans*. Esto ahorra tiempo de desarrollo, mientras otorga el poder y la flexibilidad del lenguaje Java. Y porque estos componentes hacen la mayor parte del trabajo de procesamiento, el desarrollo de componentes es más claramente separado del diseño Web, mejorando la productividad de equipos con funciones relacionadas.

3. Las JSPs son una *parte integral de la plataforma Java para la Empresa*, lo cual trae la tecnología Java a la computación de empresa. Se pueden desarrollar poderosas aplicaciones para toda la empresa, usando un sitio Web con JSP como front-end. Cuando se necesita mejorar la aplicación, se pueden mejorar los componentes y páginas Web dinámicas que residen en el servidor, manteniendo al día a todos los usuarios de una vez.

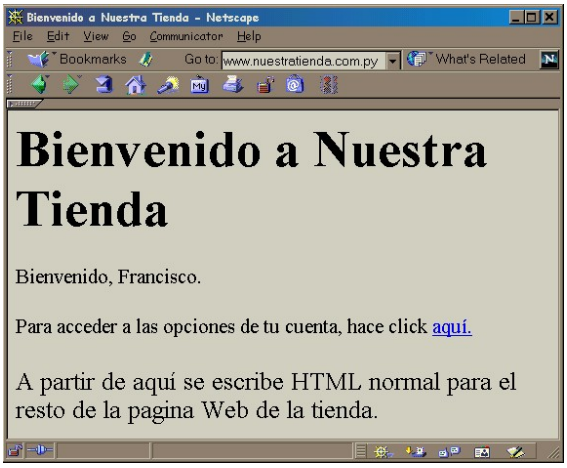
Contenido de las JSPs

Las JSPs son un tipo de archivo que combina HTML standard y nuevos tags para scripts en lenguaje Java. Uno de los nuevos tags da a las JSPs la habilidad de llamar a componentes reusables llamados JavaBeans.

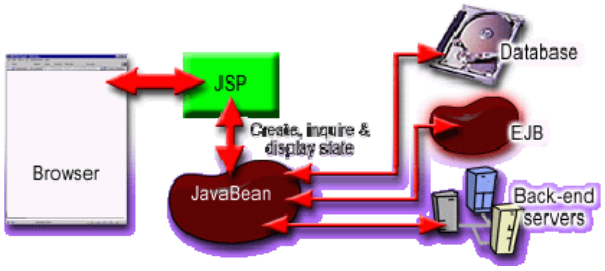
Las JSPs se parecen a HTML, pero son código que se compila a Java servlets. La responsabilidad del servlet resultante es generar una página. Esa página será una combinación del HTML del archivo JSP mezclado con contenido dinámico especificado por los nuevos tags. Así, una JSP describe una respuesta representativa de HTML. El siguiente es un ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Bienvenido a Nuestra Tienda</TITLE></HEAD>
<BODY>
<H1> Bienvenido a Nuestra Tienda </H1>
<SMALL>Bienvenido,
<!—El nombre de usuario es "Nuevo usuario" para los visitantes nuevos -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
Para acceder a las opciones de tu cuenta, hace click
<A HREF="Opciones-de-cuenta.html">aquí.</A></SMALL><P>
A partir de aquí se escribe HTML normal para el resto de la página Web de la tienda.
</BODY></HTML>
```

Esta página se ve así en un navegador:

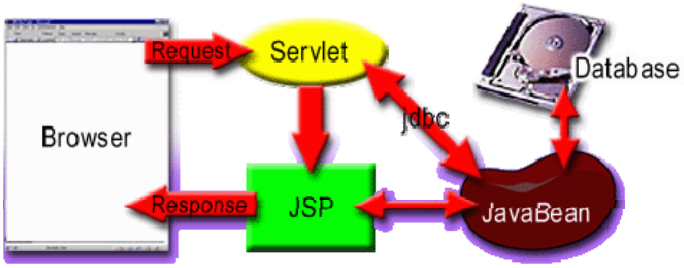


JSP Modelo 1



En este modelo, el usuario, desde un navegador, hace un pedido que es enviado a un archivo JSP. Después de recibir el pedido, el servlet que es compilado desde el archivo JSP pide información de un JavaBean. El JavaBean puede, a su vez, pedir información de un Enterprise JavaBean, una base de datos u otro servicio. Una vez que el JavaBean accede el contenido pedido, el servlet puede consultar y mostrar la información como HTML.

JSP Modelo 2



El usuario hace un pedido que es manejado por un servlet. El servlet, como en el modelo 1, pedirá contenido dinámico de un JavaBean. El contenido dinámico resultante será envuelto en un JavaBean. El servlet invocara al servlet, correspondiente a la JSP, el cual accederá al contenido dinámico del JavaBean y entregara el contenido al navegador.

Conclusión

JSP es una tecnología que facilita mucho el trabajo a los desarrolladores de sitios Web, por separar los aspectos dinámicos y estáticos de la información que desean presentar. Supera a sus similares por su portabilidad, provee una plataforma robusta de aplicaciones Web y un lenguaje y conjunto de herramientas simple y fácil de usar.

ASP – ACTIVE SERVER PAGES

Active Server Pages es un ambiente de aplicación abierta y de compilación gratuita en donde se pueden combinar HTML, scripts, y componentes reusables de un ActiveX server, para crear aplicaciones Web dinámicas y poderosas. Active Server Pages permiten programar al estilo "server side" con soporte de VBScript y Jscript.

-- *Microsoft Internet Information Server Web site*

El modo de trabajo es el siguiente: Cuando un browser hace una solicitud de un archivo .asp, el ASP es llamado por el servidor Web, que procesa el archivo solicitado desde el comienzo al final y ejecutando cualquier script que aparezca. Luego la formatea como una página Web standard y lo manda al browser.

Es posible extender los scripts **ASP** usando componentes **COM**(Common Object Model) y aún **XML**(Extensible Markup Language).

COM extiende las capacidades de programación proveyendo un método compacto, reusable y seguro para tener acceso a la información. Uno puede llamar a cualquier script o lenguaje de programación que soporte **Automation**.

XML es un lenguaje meta-markup que provee de un formato que describe datos estructurados usando una serie de tags.

ASP es escrito usualmente usando Visual Basic, Scripting Edition (**VBScript**) o Java Script (**Jscript**).

La ventaja de este modo de trabajo es que cuando se modifica un archivo ASP en el servidor, cada vez que se llama a la página, esta automáticamente se recompila. ¿Pero cual es la ventaja real ante otros lenguajes como ser CGI o Perl?. La ventaja sobre estos lenguajes es que como ASP corre como un servicio del Web Server este es más rápido y más fácil de implementar, además está optimizado para múltiples hebras (threads) y múltiples usuarios. Por otro lado, si se quiere separar el diseño de la página Web de los detalles de la programación de acceso a la base de datos, se pueden usar aplicaciones que soporten **ActiveX scripting**.

El ASP es un modelo estructurado de objetos. Pasaremos a describir algunos de los principales.

Objetos predefinidos

ASP incluye 5 objetos estándar para el uso:

- **Request**—para obtener información del usuario.
- **Response**—para mandar información al usuario.
- **Server**—para controlan el servidor de Internet.
- **Session**—para guardar la información de la sesión actual de un usuario actual.
- **Application**—para compartir y setear la información a nivel de aplicación para aquellas que tienen un determinado tiempo de vida.

Los objetos de request y response contienen ***collections*** que son bits que son accedidos de la misma manera. Los objetos contienen ***métodos*** que realizan algún tipo de acción y ***propiedades*** que contiene los atributos de un objeto.

El objeto REQUEST

Es usado para obtener información del usuario que es pasado a través de una solicitud de HTTP. Algunos métodos de este objeto son:

- **ClientCertificate:** para obtener certificación del pedido del web browser
- **QueryString:** para obtener un texto del browser
- **Form:** para obtener los datos de un form HTML
- **Cookies:** para obtener valores de un cookie de aplicación.
- **ServerVariables:** para obtener información del HTTP como el nombre del servidor.

El objeto Response

Es usado para mandar información al usuario. Algunas de sus propiedades son:

- **Buffer**—la forma de salida de la página del servidor. Cuando está en true, el servidor no mandará una respuesta hasta que todos los scripts de esta página estén procesados.
- **ContentType**—para fijar el tipo del contenido(i.e: text/HTML, Excel, etc.)
- **Expires**—fija el tiempo de expiración de la página.
- **ExpiresAbsolute**—permite fijar la fecha y hora de la expiración
- **Status**

Algunos métodos son:

- **AddHeader**—Coloca un encabezado al HTML con un valor especifico.
- **AppendToLog**—Graba en el log.
- **BinaryWrite**—escribe datos binarios (i.e, Excel)
- **Clear**—limpia el buffer del HTML.
- **End**—para de procesar los scripts.
- **Flush**—manda la información contenida en el bufer.
- **Redirect**—redirecciona al usuario a otro URL
- **Write**—escribe en el HTML, por ejemplo: Response.write("hola mundo")

El objeto Server

El objeto server es utilizado para controlar al servidor, como ser: tiempos de ocupación del servidor, creación de objetos, determinación del camino virtual de la estructura física del directorio, entre otras tareas.

El objeto Session

El objeto de sesión es usado para guardar la información de la sesión actual del usuario. Las variables definidas existirán mientras que esta sesión este activa. Soporta un método, **Abandon**, para abandonar la sesión y sus objetos. Soporta además 2 propiedades: **SessionID** y **Timeout** para la sesión

El objeto Application

Un objeto de aplicación puede guardar información que persistirá por toda la vida de una aplicación. Generalmente se usa para guardar información que existe para más de un usuario como un contador de hits. Este objeto permite manejar las hebras de una manera transparente, de donde obtiene la mayor utilidad.

Un ejemplo de programación ASP

Para poder programar en ASP se debe contar con un servidor de Web de Microsoft (IIS), se brinda el *Personal Web Server* junto con el Windows 98. Una vez instalada esta aplicación se debe grabar este ejemplo con la extensión *.asp* para crear páginas dinámicas.
En este sencillo ejemplo se muestra el poder de la programación ASP con llamadas al sistema, para hacer consultar la hora, y la estructura de control IF.

```
<%@ Language=VBScript %>
<html>
<head>
<title>Example 4</title>
</head>
<body>
<%IF Hour(time)>18 OR Hour(time)<4 THEN%>
Good Night Everyone.
<%ELSE%>
Good Morning Everyone.
<%END IF%>
</body>
</html>
```

Bibliografía

- http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/
- http://developer.netscape.com/viewsource/kuslich_jsp/kuslich_jsp.html
- http://www-4.ibm.com/software/developer/education/jsp/index.html
- http://www.ciol.com/content/Technology/Weblinks/00020701.asp
- http://msdn.microsoft.com/workshop/server/asp/ASPover.asp
- http://msdn.microsoft.com/workshop/server/asp/asptutorial.asp