

Servlet simple



Servlets

Cuando se habla de aplicaciones para Web comúnmente escuchamos hablar de JAVA, y con ello sus aplicaciones mas conocidas, los Applets, que son programas que se pueden cargar a través de una red y que se ejecutan de igual forma en cualquier plataforma, todo ello gracias a las potentes características de JAVA. Hasta hace poco, JAVA se utilizaba básicamente para dotar a las páginas WEB de una mayor interactividad mediante los Applets, y por tanto solo actuaba sobre el lado cliente. Pero el lado servidor también puede beneficiarse de todas las ventajas que ofrece JAVA, gracias a los *Servlets*.

Los *Servlets* se diferencian de los Applets básicamente en que se ejecutan en el servidor y en que no presentan ningún tipo de interfaz gráfica puesto que se encargan de hacer el trabajo oculto, un aspecto interesante por lo que muchos programadores que hasta ahora utilizaban CGI s, están utilizando Servlets. De hecho, los CGI's eran el único medio de proporcionar interacción entre el cliente y el servidor. Un ejemplo muy común de uso de los CGI's son los típicos formularios que el usuario llena con sus datos que posteriormente pasan a formar parte de una base de datos.

Los servlets son programas que funcionan como los CGI's convencionales atendiendo peticiones de un cliente teniendo al servidor como el encargado, pero escritos en Java y con la ventaja de explotar todas las bondades de java. Por ejemplo, un servlet puede ser responsable de tomar los datos de un formulario HTML y enviarlos a una base de datos para actualización de la misma.

La API Servlet, usada para escribir servlets, no incluye nada acerca de cómo son cargados los servlets, ni el ambiente en el cual corren los servlets, ni el protocolo usado para transmitir los datos del usuario. Esto permite a los servlets poder ser usados por diferentes servidores Web.

Los *Servlets* son un sustituto eficaz de los CGI s : proveen la forma de generar documentos dinámicos que son fáciles de escribir y ejecutar. También evitan el problema de desarrollar la programación según la plataforma utilizada. Los servlets son desarrollados con su propia API, una extensión estándar de Java.

¿ Porqué se afirma que los servlets podrán sustituir a los CGI's?

Al comparar a los CGI s con los servlets también nos preguntamos quién es mejor si Perl o Java. Ambos son excelentes herramientas de desarrollo y su respuesta estará dada en función de lo que se desee realizar.

Perl usa el concepto de módulos para agregar nuevas capacidades, por ejemplo el módulo DBI (Database Independent) que permite a los scripts de Perl acceder a las bases de datos (ODBC, Oracle, Sybase, Ingres, e Informix). Los módulos CGI incluyen funciones de ayuda para el desarrollo de CGI, incluyen rutinas para ser utilizadas dentro de los programas.

Java utiliza paquetes en lugar de módulos. El paquete API de servlets provee de muchas de las herramientas con las que cuentan los módulos de Perl. Para el acceso a base de datos puede usarse los respectivos drivers de conexión.

Algunas de las razones por las que se dice que los CGI's pueden ser sustituidos por servlets son :

Se programan utilizando un API estandarizada JAVA, y por tanto se benefician de todas las ventajas de este potente lenguaje.

Se pueden cargar indiferentemente y de forma transparente tanto desde un disco local como desde una dirección remota, de forma totalmente transparente.

Los Servlets pueden comunicarse entre sí, y por tanto, es posible una reasignación dinámica de la carga de proceso entre diversas máquinas. Es decir, un servlet podría pasarle trabajo a otro servlet residente en otra máquina.

Se pueden reutilizar CGI's ya hechos, incrustándolos en Servlets.

Es posible utilizarlos en servidores tan populares como el Apache, el FastTrack ó el Internet Information Server.

Los Servlets son la mejor opción a partir de ahora para desarrollar aplicaciones para servidores WEB. Trabajan de forma transparente con las filosofías GET y POST de los formularios HTML, y también, se entienden perfectamente con los Applets, pero también se pueden entender con clientes programados en cualquier otro lenguaje.

Ventajas de los servlets

Desempeño: Los servlets son mas rápidos que los CGI debido a que utilizan *threads* en lugar de *procesos*.

Una de las quejas mas comunes de los CGI/Perl es el desperdicio de los recursos del sistema. En un ambiente típico CGI, cada petición crea un nuevo proceso el cual carga el interprete de Perl. El interprete de Perl carga el script CGI, lo compila, y lo ejecuta. Además, si la aplicación se comunica con la BD, una nueva conexión será necesaria para ser hecha por cada CGI.

Los Servlets inician un nuevo thread, *hilo*, (mas que un nuevo proceso) con cada petición. Cada servlet es cargado una vez y usado más y más. Nota que a diferencia de los CGI, los servlets requieren que se cuente con una Máquina Virtual de Java (JVM) corriendo sobre el servidor todo el tiempo. Para sitios ocupados, ésta permite a los servlets usar mucho menos los recursos del sistema e incrementar el desempeño.

Portabilidad: Los servlets son tan portables como cualquier otra aplicación de Java.

Muchos sitios están buscando una solución de portabilidad. Tal vez desearían para su desarrollo usar Linux o Windows NT mientras su servidor de producción corra bajo Solaris. Ellos pueden querer vender su aplicación Web para muchos clientes como para plataformas sea posible. Hay versiones de Perl para muchas variantes de Unix como para Windows 95 y NT. Mientras que Perl para Unix no es ciento por ciento compatible con el Perl para NT, los CGIs escritos en Perl pueden ser escritos para trabajar sobre ambos. El llevar CGIs a Windows NT no es difícil de hacer, pero nunca funcionarán sin realizar al menos una modificación. Por otro lado, los CGI pueden contener comandos de Unix que no trabajarán sobre Windows NT.

La portabilidad con los servlets de Java es mas simple. Java fue diseñado para ser portable a través de todas las plataformas, permitiendo que las aplicaciones sean movidas fácilmente de un sistema operativo a otro.

Seguridad

La seguridad puede ser un problema cuando se desarrollan CGIs. El área de mas preocupación es el proceso de estrada del usuario. Este podría ser desde formas o desde datos dentro de un URL. Muchos CGIs escritos en Perl son vulnerables a ataques donde la finalidad del usuario es trucar el CGI ejecutando algún comando sobre el servidor. Los Servlets no están en riesgo de correr comando de shell no planeados. Los lenguajes compilados como Java (o C) proveen mejor seguridad que los lenguajes que interpretan scripts.

Los Servlets son archivos de clases compilados mientras que un CGI/Perl es manipulado en su forma de código fuente. Dependiendo quien tenga acceso al servidor Web, se puede elegir entre instalar o no el código fuente.

Desarrollo

El desarrollo con Perl es simple. Lo único que se necesita es un editor y mucha documentación de Perl.. Mientras que a muchos desarrolladores les agrada este ambiente, hay una creciente proliferación de herramientas IDE (Integrated Development Environment) para desarrollo. En ese campo, Java por encima de Perl. Hay IDEs disponibles desde Sun, Microsoft, Symantec y muchas otras, proporcionando simples entornos GUI. Las IDEs pueden proveer útiles herramientas fáciles de aprender para nuevos desarrolladores.

Compartiendo el trabajo entre los desarrolladores es además es más fácil con Java, o puede asignarse en paquetes individuales. Cuando se trabaja con una aplicación CGI/Perl donde dos o mas desarrolladores estuvieron trabajando simultáneamente, la aplicación normalmente es manejada de un desarrollador a otro. El desarrollo de aplicaciones Java es mucho mejor para proyectos largos.

Arquitectura de los Servlets

El principal componente de la Servlet API es la interfaz Servlet. Todos los servlets implementan esta interfaz directamente, por medio extensión de la clase que la implementa, HttpServlet. Esta interfaz está provista de métodos que manipulan a los servlets y la comunicación con sus clientes.

Cuando un servlet es llamado desde un cliente, este recibe dos objetos: ServletRequest y ServletResponse. La interfaz ServletRequest se encarga la comunicación desde el cliente al servidor, mientras que la interfaz ServletResponse atiende la comunicación desde servlet al cliente.

La interfaz ServletRequest permite al servlet acceder a información como, los nombres de parámetros pasados por el cliente, el protocolo usado por el cliente, y los nombres de los host remotos que hacen la solicitud y el servidor que la recibe. Est interfaz permite a los servlets el acceso a métodos que permiten manejar la presentación de la respuesta como salida en el navegador, a través de los cuales consiguen los datos desde el cliente que usa protocolos como HTTP POST , etc..

La interfaz ServletResponse proporciona al servlet los métodos para contestarle al cliente. Permite al servlet configurar la forma de salida de los datos para el cliente, ServletOutputStream que permite enviar la replica de datos como respuesta. La subclases de ServletResponse le dan mas capacidad al servlet para responder.

Las clases e interfaces descritas conforman a un servlet básico. Pero existen métodos adicionales que provee la API con la capacidad para controlar sesiones o múltiples conexiones, entre muchas más aplicaciones

Ejemplos de aplicación de los servlets

Algunas de las aplicaciones de servlets incluyen :

Procesamiento de datos enviados con HTTPS a partir de una forma HTML, incluyendo la orden de compra o datos de una tarjeta de crédito. Un servlet como este podría ser parte de una orden de entrada al sistema de procesamiento de datos, haciendo la respectiva actualización en la base de datos, y quizá hasta un sistema de pago en línea.

Interacción múltiple entre personas. Un servlet puede manipular múltiples peticiones al mismo tiempo; podrían sincronizar dichas solicitudes de sistemas que dan soporte, por ejemplo conferencias en línea.

Redireccionamiento de peticiones. Los servlets pueden reenviar las peticiones a otros servidores y servlets. Esto permite que al ser usados se balancee la carga entre varios servidores que atiendan al misma tarea, según el tipo de proceso que deban atender.

Aplicaciones más específicas implementadas con servlets

Con los Servlets se puede implementar por ejemplo:

Típicos sistemas middleware que hasta ahora únicamente se implementaban con CGI's. Para consultar las bases de batos, los Servlets pueden utilizar JDBC (Java Data Base Connection), lo que les permite extraer información de cualquier sistema de Base de Datos.

Automatización de un sistema de recepción y publicación de información. Por ejemplo podríamos montarnos una simple estación meteorológica que permitiese acceso a su información mediante una página WEB. Por un lado tendríamos un servlet que recolectaría la información de los diversos tipos de sensores y la almacenaría en bases de datos, y por otro lado, un servlet que se encargaría de presentar esta información en función de las peticiones del cliente basándose en estas mismas bases de datos.

Control de la recepción de correo electrónico, y de sistemas de news, chats, etc. Conviene recordar que Java está especialmente indicado para la programación utilizando los protocolos TCP/IP.

Dado que pueden manejar múltiples peticiones en forma concurrente, es posible implementar aplicaciones de colaboración, como por ejemplo una aplicación de videoconferencia.

