

Programación orientada a objetos

Práctica 8: SERVLETS



Alumno: Vargas Romero Erick Efraín

Profesor: Tecla Parra Roberto

Fecha: 08-06-2017

Grupo: 2CM4

**Introducción**

Para é uso de servlets se debe implementar un servidor usando “TomCat”, el cual fuera contendor del servlet, para con ello, poder ejecutarlo desde nuestro equipo. Cabe mencionar, que un servlet no es más que un programa que funciona del lado del servidor, en contraste con un applet que se ejecuta del lado del equipo cliente. Un applet es ejecutado por el navegador, mientras un servlet lo ejecuta un programa de tipo servidor como “TomCat”, en este caso.

**Desarrollo**

En esta práctica se hace uso de dos clases en específico, la primera jamada Tabla, la cual es una clase que extiende de una clase HttpServlet y tiene para iniciar el método protegido llamado service, el cual recibe dos argumentos, el primero es un HttpServletRequest, el segundo un HttpServletResponse además de que puede lanzar excepciones tipo ServletException o bien IOException, siendo así, primeramente nuestro objeto tipo HttpServletResponse le añadimos el tipo de contenido, en nuestro caso se usa texto html, posteriormente se crea un objeto PrintWriter, el cual nos ayudará a “escribir” HTML.

Además, tenemos una variable llamada num, la cual obtiene un parámetro de tipo numérico, el cual nos ayudará a evaluar que tabla de multiplicar vamos a realizar.

Una vez hecho lo anterior solo resta mostrar la tabla de multiplicar, esto se hace usando un ciclo for.

public class Tabla **extends** HttpServlet **{**

protected void service**(**HttpServletRequest request**,** HttpServletResponse response**)**

**throws** ServletException**,** IOException **{**

response**.**setContentType**(**"text/html;Charset=UTF-8"**);**

PrintWriter salida **=** response**.**getWriter**();**

int num **=** Integer**.**parseInt**(**request**.**getParameter**(**"cmpnum"**));**

salida**.**println**(**"<html><head><title>Tabla de multiplicar</title></head>"**);**

salida**.**println**(**"<body>"**);**

salida**.**println**(**"<center><h2>Tabla de multiplicar del "**+**num**+**"</h2></center>"**);**

salida**.**println**(**"<table align='center' border='1'>"**);**

**for(**int i**=**1**;** i**<=**10**;** i**++){**

salida**.**println**(**"<tr><td bgcolor='white'>"**+**num**+**" x "**+**i**+**"</td>"**);**

salida**.**println**(**"<td bgcolor='gray'>"**+(**num**\***i**)+**"</td>"**);**

**}**

salida**.**println**(**"</table>"**);**

salida**.**println**(**"<br/><br/><center><a href='http://localhost:8080/prog9/'>Volver</a></center>"**);**

salida**.**println**(**"</body></html>"**);**

salida**.**close**();**

**}**

**}**

Tenemos otra clase llamada tablas, la cual extiende de una clase HttpServlet, y tiene un método protegido llamado doPost el cual recibe dos argumentos el primero es un objeto tipo HttpServletRequest, y el otro HttpServletResponse, y de igual manera que el método de la clase ya mencionada nos puede arrojar excepciones tipo ServletException o bien IOException, nuevamente establecemos el tipo de contenido, posteriormente, el objeto con el cual escribiremos en nuestro HTML, siendo así evaluamos si el número que se ha obtenido desde un archivo HTML es mayor a cero pero menor o igual a 12 si es así, se procede a enviar a la clase previamente mencionada y mostrar las tablas de multiplicar sino se manda un mensaje de que no existe esa tabla de multiplicar.

**import** java**.**io**.**IOException**;**

**import** java**.**io**.**PrintWriter**;**

**import** javax**.**servlet**.**ServletException**;**

**import** javax**.**servlet**.**annotation**.**WebServlet**;**

**import** javax**.**servlet**.**http**.**HttpServlet**;**

**import** javax**.**servlet**.**http**.**HttpServletRequest**;**

**import** javax**.**servlet**.**http**.**HttpServletResponse**;**

@WebServlet**(**urlPatterns **=** **{**"/Tablas"**})**

public class Tablas **extends** HttpServlet **{**

protected void doPost**(**HttpServletRequest request**,** HttpServletResponse response**)**

**throws** ServletException**,** IOException **{**

response**.**setContentType**(**"text/html;Charset=UTF-8"**);**

PrintWriter salida **=** response**.**getWriter**();**

int num **=** Integer**.**parseInt**(**request**.**getParameter**(**"cmpnum"**));**

**if(**num**>**0 **&&** num**<=**12**){**

response**.**sendRedirect**(**"Tabla?cmpnum="**+**num**);**

**}else{**

salida**.**println**(**"<html><body>"**);**

salida**.**println**(**"<center><h2><font color='red'>No existe esta tabla de multiplicar</font></h2></center>"**);**

salida**.**println**(**"<br/><br/><center><a href='http://localhost:8080/prog9/'>Volver</a></center>"**);**

salida**.**println**(**"</body></html>"**);**

salida**.**close**();**

**}**

**}**

**}**

**Conclusión**

Gracias al desarrollo de esta práctica se aprendió a implementar tanto un servidor, un contenedor de servlets y un servlet en lenguaje Java. Cabe destacar que combinar código html con java fue lo que determinó la creación del programa, en mi opinión, es una tarea ardua para el programador tener que crear su código en html y enseguida incrustar cada etiqueta en su correspondiente sentencia en java. Por lo que el uso de servlets yo lo utilizaría para crear un formulario, así, existiría una sección que reciba información y la envíe a una base de datos, esa única parte sería destinada para interactuar con el servidor propiamente, tal y como lo hace php, por citar un ejemplo.