



Facultad de Ingeniería en Sistemas de Información y Ciencias de la Computación

Carrera: 090-Ingeniería en Sistemas - Docente: Carlos Alejandro Arias López

Curso: Web Developer - Código del curso: 036 – Laboratorio 5

Sección: A - Ciclo: 8to - Horario: 7:00 am a 8:35 - Días: lunes, miércoles y viernes

Servlets Java: pasos para crear Servlet

1. Crear una estructura de directorio
2. Crear un servlet
3. Compila el servlet
4. Agregar asignaciones al archivo web.xml
5. Inicie el servidor e implemente el proyecto.
6. Acceda al servlet

Siga la guía como apoyo. Recuerde que la versión con la cual fue creado este proyecto fue en versión 8 de Netbeans corrija los pasos y cree un breve informe de su trabajo. Enseñe su trabajo al docente.

Comprendamos cómo funciona un servlet.

Para ejecutar un programa de servlet, debemos tener Apache Tomcat Server instalado y configurado. *Eclipse para Java EE proporciona Apache Tomcat incorporado.* Una vez que el servidor está configurado, puede comenzar con su programa. Un punto importante a tener en cuenta: para cualquier programa de servlet, necesita 3 archivos: *archivo index.html*, *archivo de clase Java* y *archivo web.xml*. El primer paso es crear un proyecto web dinámico y luego continuar.

Ahora, veamos cómo agregar 2 números usando servlets y mostrar la salida en el navegador.

Primero, escribiré el archivo index.html



```
<!DOCTYPE HTML>
<html>
<body>

<form action = "add">
Enter 1st number: <input type="text" name ="num1">
  Enter 2nd number: <input type="text" name="num2">
</form>

</body>
</html>
```

El programa anterior crea un formulario para ingresar los números para la operación de suma. Sin el archivo de clase Java, no puede realizar la suma de 2 números. Así que ahora creemos un archivo de [clase](#) .

Después de escribir el archivo de clase Java, el último paso es agregar asignaciones al archivo web.xml. Veamos cómo hacer eso.

```
package edureka;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Add extends HttpServlet{
public void service(HttpServletRequest req, HttpServletResponse
res) throws IOException
{
int i = Integer.parseInt(req.getParameter("num1"));
int j = Integer.parseInt(req.getParameter("num2"));
int k= i+j;
PrintWriter out = res.getWriter();
out.println("Result is"+k);
}
}
```

El archivo *web.xml* estará presente en la carpeta WEB-INF de su contenido web. Si no está presente, puede hacer clic en Descriptor de implementación y hacer clic en *Generar código auxiliar de descriptor de implementación*. Una vez que tenga listo su archivo web.xml, debe agregarle las asignaciones. Veamos cómo se realiza el mapeo utilizando el siguiente ejemplo:



Una vez hecho esto, puede ejecutar el programa iniciando el servidor y obtener la salida deseada en el navegador.

Tomemos otro ejemplo donde crearé un servlet de inicio de sesión simple. Nuevamente, el primer paso será escribir un archivo html.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0">
<display-name>Basic</display-name>
<servlet>
<servlet-name>Addition</servlet-name>
<servlet-class>edureka.Add</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Addition</servlet-name>
<url-pattern>/add</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```



```
<!DOCTYPE html>
<html>
<body>
<form action="Login" method="post">
<table>
<tr>
<td><font face="Noto Serif" size="2px">Name:</font></td>
<td><input type="text" name="userName"></td>
</tr>
<tr>
<td><font face="Noto Serif" size="2px">Password:</font></td>
<td><input type="password" name="userPassword"></td>
</tr>
</table>
<input type="submit" value="Login">
</form>
</body>
</html>
```

A continuación, codifiquemos el archivo Java Class.

```
package Edureka;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Login extends HttpServlet
{
protected void doPost(HttpServletRequest req,HttpServletResponse
res)throws ServletException,IOException
{
PrintWriter pw=res.getWriter();
res.setContentType("text/html");
String user=req.getParameter("userName");
String pass=req.getParameter("userPassword");
pw.println("Login Success...!");
if(user.equals("edureka") && pass.equals("edureka"))
pw.println("Login Success...!");
else
pw.println("Login Failed...!");
pw.close();
}
}
```



En el código anterior, he establecido una condición: si el nombre de usuario y la contraseña son iguales a **edureka**, solo entonces se mostrará correctamente conectado, de lo contrario, se denegará el inicio de sesión.

Agreguemos las asignaciones al archivo web.xml ahora.

```
<?xml version="1.0"encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"version="3.1">
<display-name>LoginServlet</display-name>
<servlet>
<servlet-name>Login</servlet-name>
<servlet-class>Edureka.Login</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Login</servlet-name>
<url-pattern>/Login</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```

Fuente: <https://www.edureka.co/blog/java-servlets>