**EXPERIMENT NO: 5**

**a. Write a program for demonstration of Generic Servlets.**

**Theory**

**GenericServlet** class implements **Servlet**, **ServletConfig** and **Serializable** interfaces. It provides the implementaion of all the methods of these interfaces except the service method.

GenericServlet class can handle any type of request so it is protocol-independent.

You may create a generic servlet by inheriting the GenericServlet class and providing the implementation of the service method.

### Methods of GenericServlet class

There are many methods in GenericServlet class. They are as follows:

1. **public void init(ServletConfig config)** is used to initialize the servlet.
2. **public abstract void service(ServletRequest request, ServletResponse response)** provides service for the incoming request. It is invoked at each time when user requests for a servlet.
3. **public void destroy()** is invoked only once throughout the life cycle and indicates that servlet is being destroyed.
4. **public ServletConfig getServletConfig()** returns the object of ServletConfig.
5. **public String getServletInfo()** returns information about servlet such as writer, copyright, version etc.
6. **public void init()** it is a convenient method for the servlet programmers, now there is no need to call super.init(config)
7. **public ServletContext getServletContext()** returns the object of ServletContext.
8. **public String getInitParameter(String name)** returns the parameter value for the given parameter name.
9. **public Enumeration getInitParameterNames()** returns all the parameters defined in the web.xml file.
10. **public String getServletName()** returns the name of the servlet object.
11. **public void log(String msg)** writes the given message in the servlet log file.
12. **public void log(String msg,Throwable t)** writes the explanatory message in the servlet log file and a stack trace.

**Example on GenericServlet**

**Develop a program to authenticate the user name and password**

//Client form

<html>

<head>

</head>

<body>

<form name=form1 method=post action="http://localhost:8080/mysite/servlet/GenericServletDemo2">

Username: <input type=text name=user1 value=""><br>

Password: <input type=text name=password1 value=""><br>

<input type=submit value="login">

<input type=reset value="reset">

</form>

</body>

</html>

//servlet

import java.io.\*;

import javax.servlet.\*;

public class GenericServletDemo2 extends GenericServlet

{

public void service(ServletRequest request ,ServletResponse response)

throws ServletException,IOException

{

String uname="Prasad";

String pass="123456";

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<HTML>");

out.println("<HEAD><TITLE>hello page</TITLE><HEAD>");

out.println("<BODY>");

String name=(String) request.getParameter("user1");

String password=(String) request.getParameter("password1");

if((name.equals(uname))&&(password.equals(pass)))

{

out.println("Welcome "+name);

}

else

{

out.println("Invalid username or password");

}

out.println("</BODY></HTML>");

}

}

Output:



When Valid user



When Invalid User



**b. Write a program for demonstration of HTTP Servlets.**

# HttpServlet class

|  |
| --- |
| The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead, doTrace etc. |

### Methods of HttpServlet class

There are many methods in HttpServlet class. They are as follows:

1. **public void service(ServletRequest req,ServletResponse res)** dispatches the request to the protected service method by converting the request and response object into http type.
2. **protected void service(HttpServletRequest req, HttpServletResponse res)** receives the request from the service method, and dispatches the request to the doXXX() method depending on the incoming http request type.
3. **protected void doGet(HttpServletRequest req, HttpServletResponse res)** handles the GET request. It is invoked by the web container.
4. **protected void doPost(HttpServletRequest req, HttpServletResponse res)** handles the POST request. It is invoked by the web container.
5. **protected void doHead(HttpServletRequest req, HttpServletResponse res)** handles the HEAD request. It is invoked by the web container.
6. **protected void doOptions(HttpServletRequest req, HttpServletResponse res)** handles the OPTIONS request. It is invoked by the web container.
7. **protected void doPut(HttpServletRequest req, HttpServletResponse res)** handles the PUT request. It is invoked by the web container.
8. **protected void doTrace(HttpServletRequest req, HttpServletResponse res)** handles the TRACE request. It is invoked by the web container.
9. **protected void doDelete(HttpServletRequest req, HttpServletResponse res)** handles the DELETE request. It is invoked by the web container.
10. **protected long getLastModified(HttpServletRequest req)** returns the time when HttpServletRequest was last modified since midnight January 1, 1970 GMT.

**Example**

**Develop a program to retrieve user’s data**

**HTML File - GetUserData.html**

<html>

<body>

<center>

<form name="Form1" method=get action="http://localhost:8080/mysite/servlet/ GetUserData">

<B>Student Details:</B>

<BR/>

<table>

<tr>

<td><B>Student Roll No.:</td>

<td><input type=textbox name="roll" size="25" value=""></td>

</tr>

<tr>

<td><B>Student Name:</td>

<td><input type=textbox name="name" size="25" value=""></td>

</tr>

</table>

<input type=submit value="Submit">

</form>

</body>

</html>

**Servlet File - GetUserData.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class GetUserData extends HttpServlet

{

public void doGet(HttpServletRequest request, HttpServletResponse response)throws IOException, ServletException

 {

 String roll = request.getParameter("roll");

 String name = request.getParameter("name");

 response.setContentType("text/html");

 PrintWriter pw = response.getWriter();

 pw.println("<B>Student Roll No.: ");

 pw.println(roll);

 pw.println("<BR>");

 pw.println("<B>Student Name: ");

 pw.println(name);

 pw.close();

 }

}

**Output:**



