

LAB EXPERIMENT : 02

Write a Servlet program that prints "Hello, World!" when accessed through a browser.

Aim:

To write a simple Servlet program in Java that prints "Hello, World!" when accessed through a browser.

Algorithm:

1. Set up the Servlet Environment:

- Install a Servlet container like **Apache Tomcat**.
- Set up the Java Development Kit (JDK) and Java Enterprise Edition (JEE) to compile and run the Servlet.

2. Create a Java Servlet Class:

- Create a class that extends `HttpServlet`.
- Override the `doGet()` method to respond to HTTP GET requests.

- Inside the doGet() method, use PrintWriter to send the response "Hello, World!" back to the browser.

3. Configure the Servlet in web.xml:

- Add a <servlet> entry and <servlet-mapping> entry in the web.xml file to map the Servlet class to a URL.

4. Deploy and Test the Servlet:

- Deploy the Servlet in the Tomcat server.
- Access the URL of the servlet through a browser to see the output "Hello, World!".

Code :

Java Servlet Code:

```
import java.io.*;
```

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
public class HelloWorldServlet extends HttpServlet {
```

```
protected void doGet(HttpServletRequest request,  
HttpServletResponse response) throws  
ServletException, IOException {
```

```
    // Set the content type of the response to text/html  
    response.setContentType("text/html");
```

```
    // Get the PrintWriter object to send response  
    PrintWriter out = response.getWriter();
```

```
    // Write the HTML content in the response  
    out.println("<html><body>");  
    out.println("<h1>Hello, World!</h1>");  
    out.println("</body></html>");  
}
```

```
}
```

web.xml Configuration:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"
```

```
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```
http://java.sun.com/xml/ns/javaee/web-app_3_1.xsd"
```

```
    version="3.1">
```

```
<servlet>
```

```
    <servlet-name>HelloWorldServlet</servlet-name>
```

```
    <servlet-class>HelloWorldServlet</servlet-class>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
    <servlet-name>HelloWorldServlet</servlet-name>
```

```
    <url-pattern>/hello</url-pattern>
```

```
</servlet-mapping>
```

</web-app>

Steps to Run the Servlet:

1. Set up the Environment:

- Install Apache Tomcat or any other servlet container.
- Configure the JDK and set up your IDE (e.g., Eclipse or IntelliJ IDEA) to support Java web development.

2. Compile the Servlet:

- Save the HelloWorldServlet.java file in the appropriate directory.
- Use the javac command to compile the Java file or let your IDE handle this for you.

3. Deploy the Servlet:

- Place the compiled class in the WEB-INF/classes directory of your web application.
- Make sure web.xml is placed in the WEB-INF folder.

4. Start Tomcat Server:

- Start the Apache Tomcat server.
- Open the browser and access the URL `http://localhost:8080/your-app/hello`.

OUTPUT:



RESULT :

The servlet successfully prints "Hello, World!" when accessed via the browser.

This demonstrates a basic Java Servlet and how it can be used to generate dynamic web content in response to user requests.