**URL Rewriting using Java Servlet**

Url rewriting is a process of appending or modifying any url structure while loading a page.

The request made by client is always a new request and the server can not identify whether the current request is send by a new client or the previous same client. Due to This property of HTTP protocol and Web Servers are called stateless. But many times we should know who is client in the processing request.  
**For example:**  
In any social networking site during login to till logout the server should know who is client so that server can manage all the request according to the user need.  
This problem is solved by Session in Servlet.

**Session :** Session is a state between client and server and it contain multiple request and response between client and server. As we know that HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

**Following are some ways by which we can provide unique id in request and response :**

 [Session Interface in Servlet](https://www.geeksforgeeks.org/the-httpsession-interface-in-servlet/)

 [Cookies Management](https://www.geeksforgeeks.org/cookies-used-website/)

 URL Rewriting

**URL Rewriting**

If your browser does not support cookies, URL rewriting provides you with another session tracking alternative. URL rewriting is a method in which the requested URL is modified to include a session ID. There are several ways to perform URL rewriting.  
Here we are discussing session management technique in which we pass control from one servlet to another. The Syntax for passing control from one servlet to another is as follows

|  |
| --- |
| // SecondServlet is the name of the servlet where you want to pass control  out.print("<a href='SecondServlet?uname=" + n + "'>visit</a>"); |

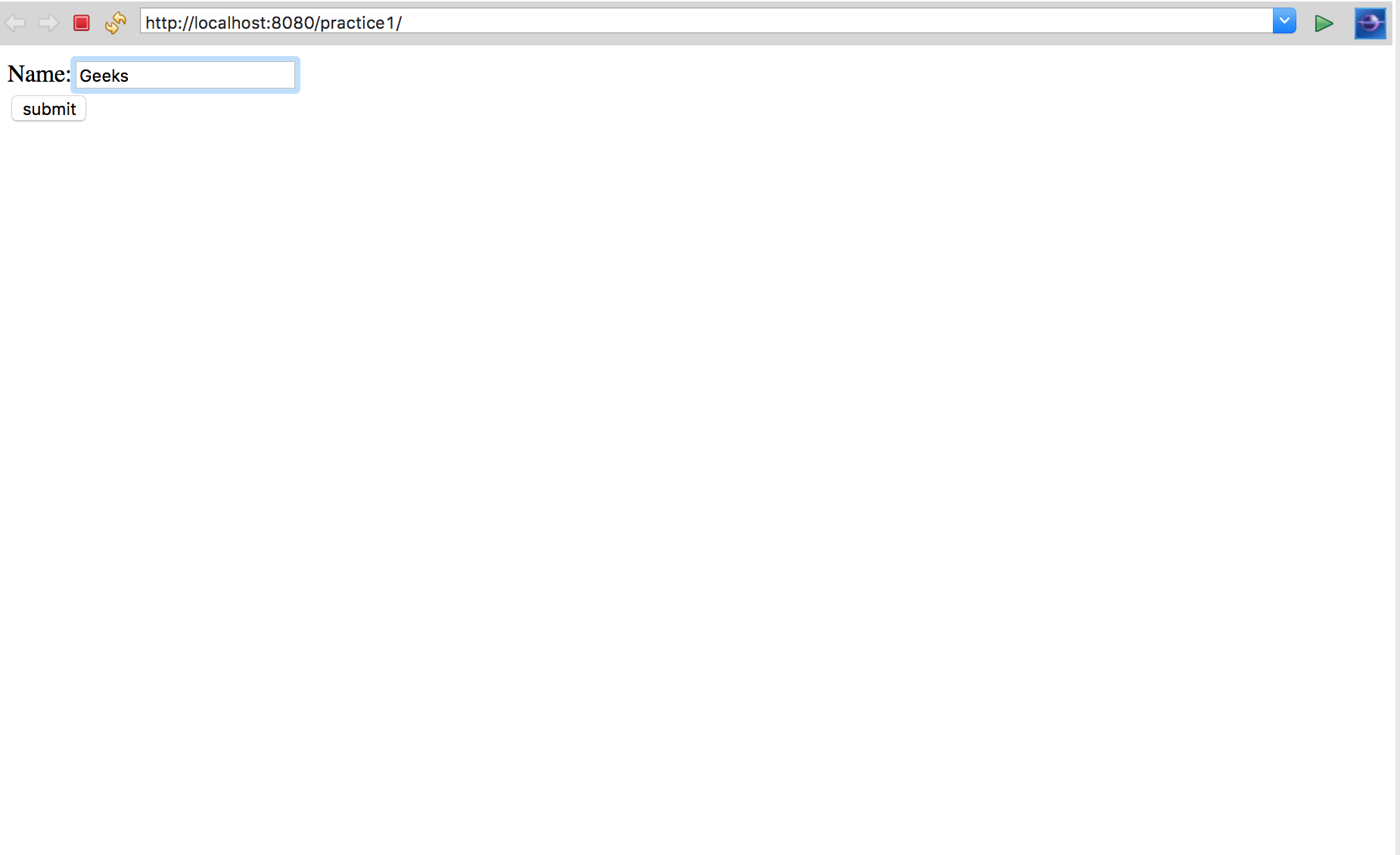
**Note :** Generally we write web.xml file for request dispatcher but in this example we use annotation so their is no need of creating web.xml file.

Syntax of annotation is :

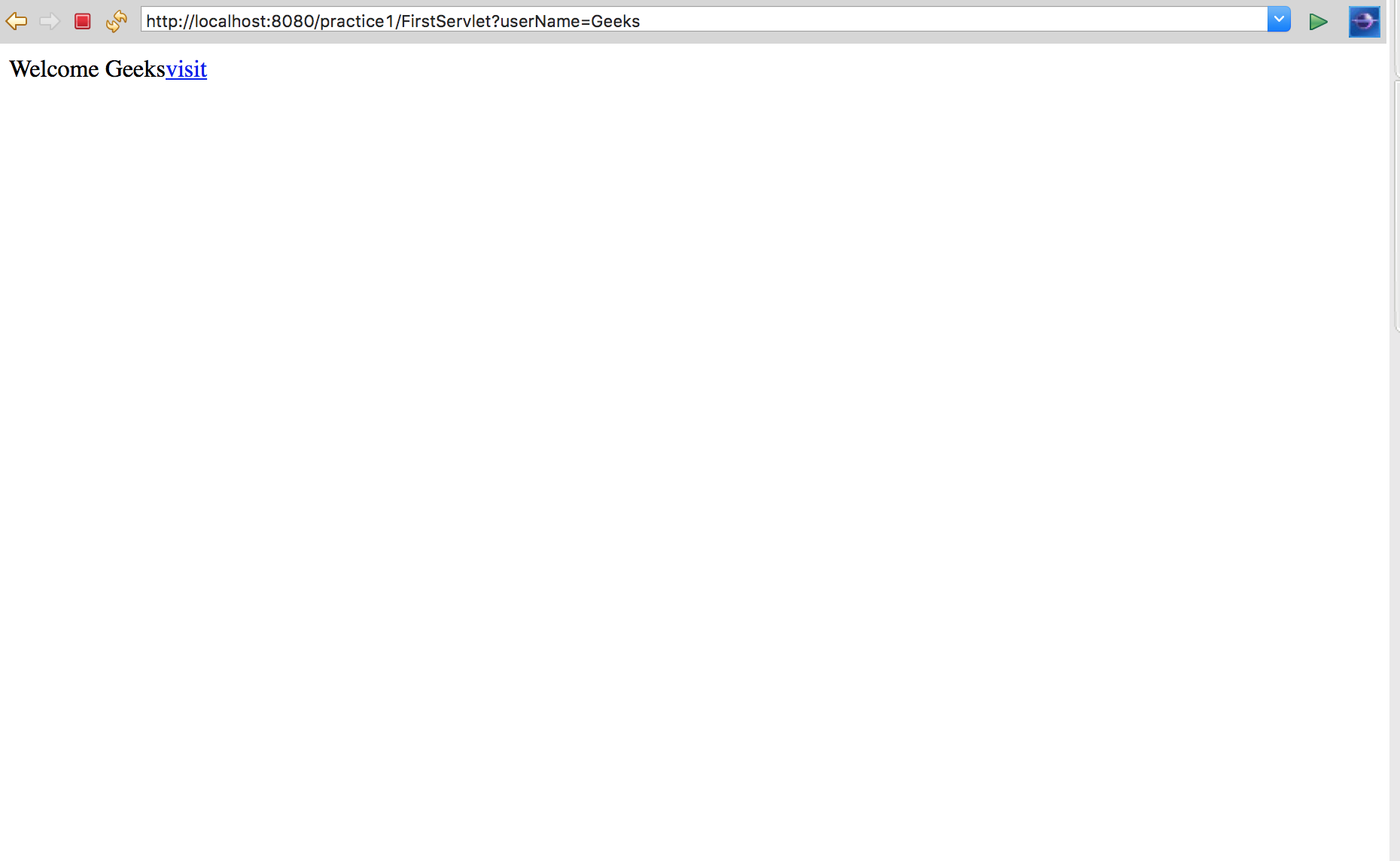
|  |
| --- |
| @WebServlet("/ServletName") // at the place of ServletName  we have to write the name of our servlet file and **this** annotation  should be written at the starting of **class** in servlet. |

**Example of Session tracking using URL rewriting using annotation**

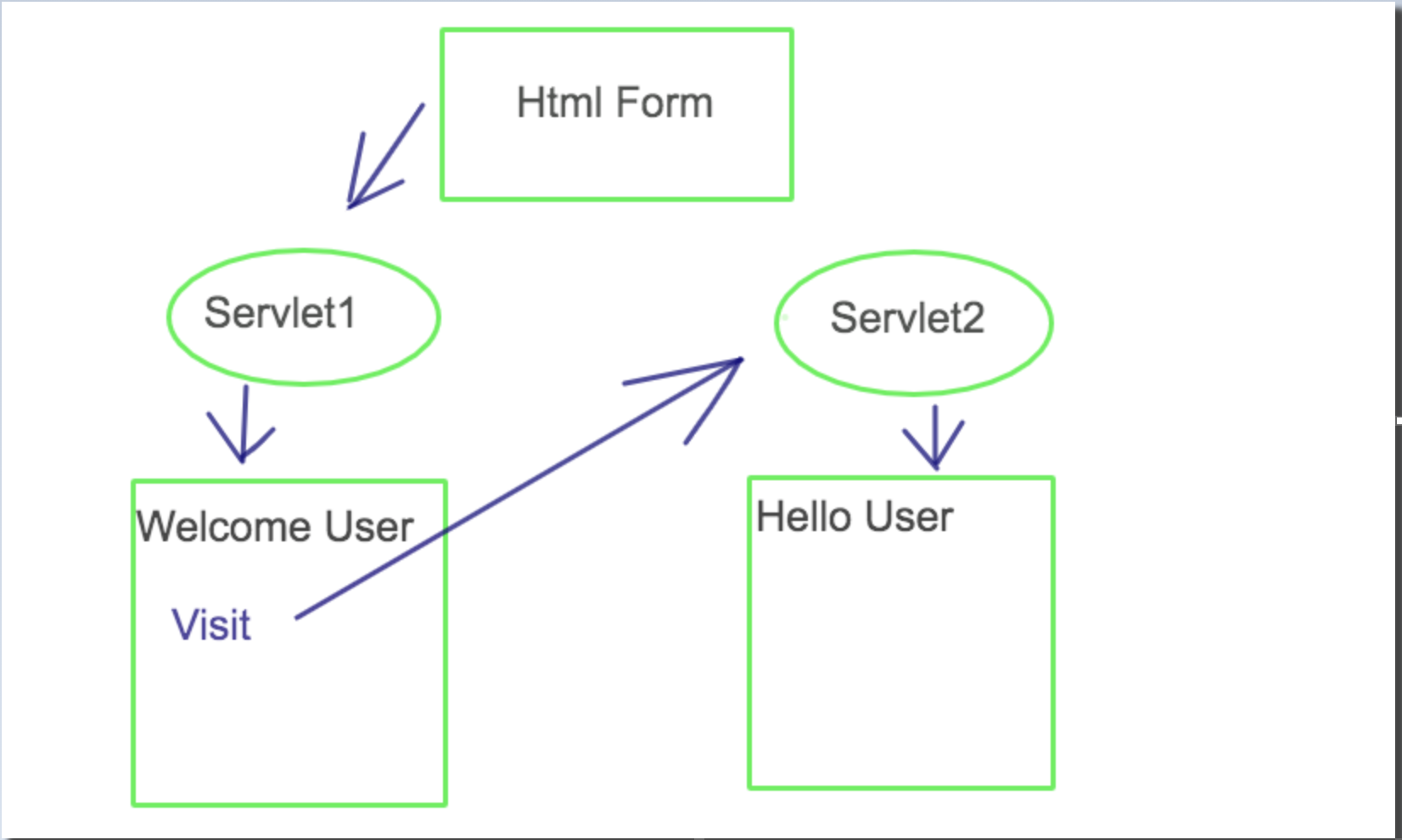
|  |
| --- |
| <!-- Save this file as Index.html -->  <!DOCTYPE html>  <**html**>  <**head**>  <**meta** charset="UTF-8">  <**title**>Insert title here</**title**>  </**head**>  <**body**>  <**form** action="FirstServlet" method="get">  Name:<**input** type="text" name="userName"/><**br**/>  <**input** type="submit" value="submit"/>  </**form**>  </**body**>  </**html**> |



|  |
| --- |
| // Name this file as FirstServlet.java  **package** GeeksforGeeks;  **import** java.io.\*;  **import** javax.servlet.\*;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.\*;  @WebServlet("/FirstServlet") // annotation    // this annotation is used for replacing xml file  **public** **class** FirstServlet **extends** HttpServlet {        // class name is FirstServlet which extends HttpServlet  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)      {  **try** {              response.setContentType("text/html");             /\* The response's character encoding is only set  from the given content type if this method  is called before getWriter is called.  This method may be called repeatedly to  change content type and character encoding.\*/              PrintWriter out = response.getWriter();                  /\*T he Java PrintWriter class ( java.io.PrintWriter ) enables you to            write formatted data to an underlying Writer .           For instance, writing int, long and other primitive data            formatted as text, rather than as their byte values\*/              String n = request.getParameter("userName");    //request.getParameter takes the value from index.html file    // where name is username              out.print("Welcome " + n);    // out.println is used to print on the client web browser     //url rewriting is used for creating session  //       it will redirect  you to SecondServlet page              out.print("<a href='SecondServlet?uname=" + n + "'>visit</a>");                out.close();          }  **catch** (Exception e) {              System.out.println(e);          }      }  } |



|  |
| --- |
| // Name this file as SecondServlet.java  **package** GeeksforGeeks;    **import** java.io.\*;  **import** javax.servlet.\*;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.\*;    @WebServlet("/SecondServlet") // annotation  **public** **class** SecondServlet **extends** HttpServlet {    **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)      {  **try** {                response.setContentType("text/html");              /\*               \* The response's character encoding is only set from the given               \* content type if this method is called before getWriter is called.               \* This method may be called repeatedly to change content type and               \* character encoding.               \*/              PrintWriter out = response.getWriter();              /\*               \* The Java PrintWriter class ( java.io.PrintWriter ) enables you to               \* write formatted data to an underlying Writer . For instance,               \* writing int, long and other primitive data formatted as text,               \* rather than as their byte values               \*/              // getting value from the query string              String n = request.getParameter("uname");              out.print("Hello " + n);              /\* out.println is used to print on the client web browser \*/              out.close();          }  **catch** (Exception e) {              System.out.println(e);          }      }  } |

  
**Program flow**  


**Explanation(Follow the diagram)**  
When you deploy your project in eclipse the first page which is loaded in the HTML form whose form action is first servlet so the control will go to servlet1. In this case, we name servlet1 as FirstServlet where the username is printed. In FirstServlet we provide url where we transfer the control to servlet2 using url rewriting. In our case we name servlet2 as SecondServlet.  
**Advantage of URL Rewriting :**

* It doesn’t depend upon cookies and will work whether cookies are enabled or disabled
* Extra form submission is not required on all pages

**Disadvantage of URL Rewriting**  
It will work only with links ad can send only text