**Session tracking using Hidden Form Field using Annotation | Java Servlet**

Hidden form field is used to store session information of a client. In this method, we create a hidden form which passes the control to the servlet whose path is given in the form action area. Using this, the information of the user is stored and passed to the location where we want to send data.   
The main advantage of using Hidden form filed that it doesn’t depend on the browser. Even If the cookies are disabled or not hidden, form filed will work perfectly.

**Example**

|  |
| --- |
| out.print("<form action='SecondServlet'>");  out.print("<input type='hidden' name='username'  value='" + n + "'>");  out.print("<input type='submit' value='submit'>");  out.print("</form>"); |

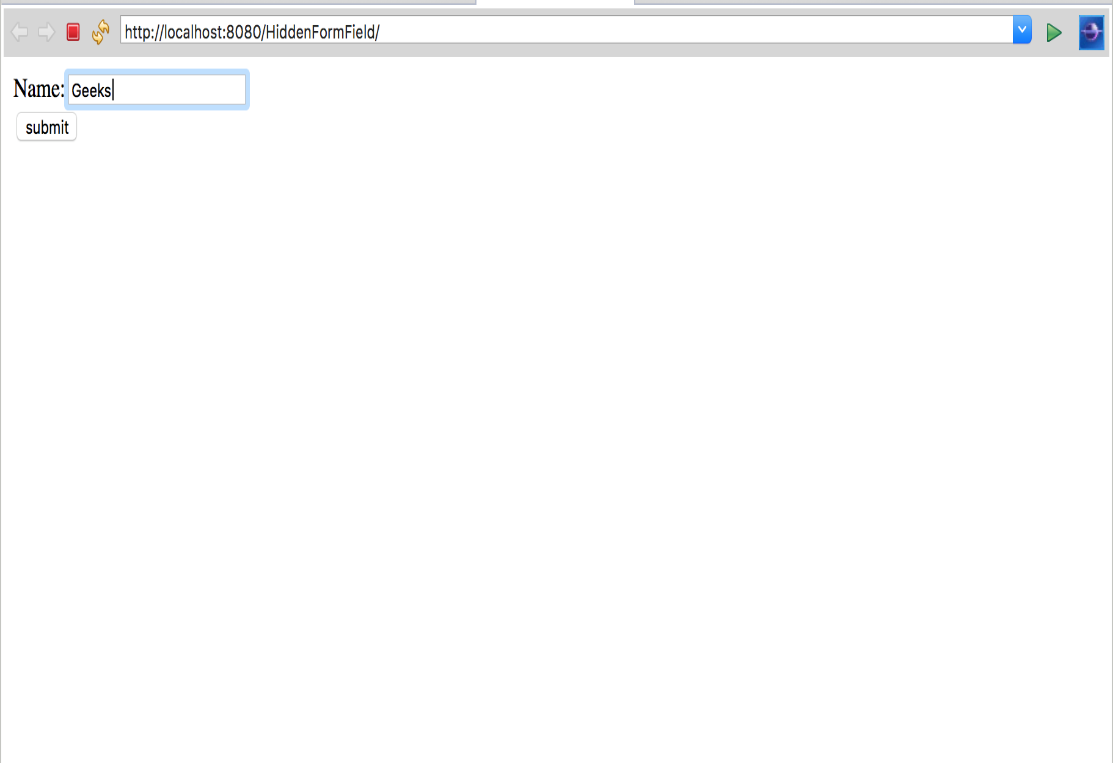
**A complete example of Hidden form field method**

In this method we are transferring the client detail from firstServlet to Second Servlet

**Index.html**

|  |
| --- |
| <!DOCTYPE html>  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Insert title here</title>  </head>  <body>      <form action="FirstServlet" method="Post">          <!-- Move the control to firstServlet -->          Name:<input type="text" name="userName" /><br />          <input type="submit" value="submit" />      </form>  </body>  </html> |

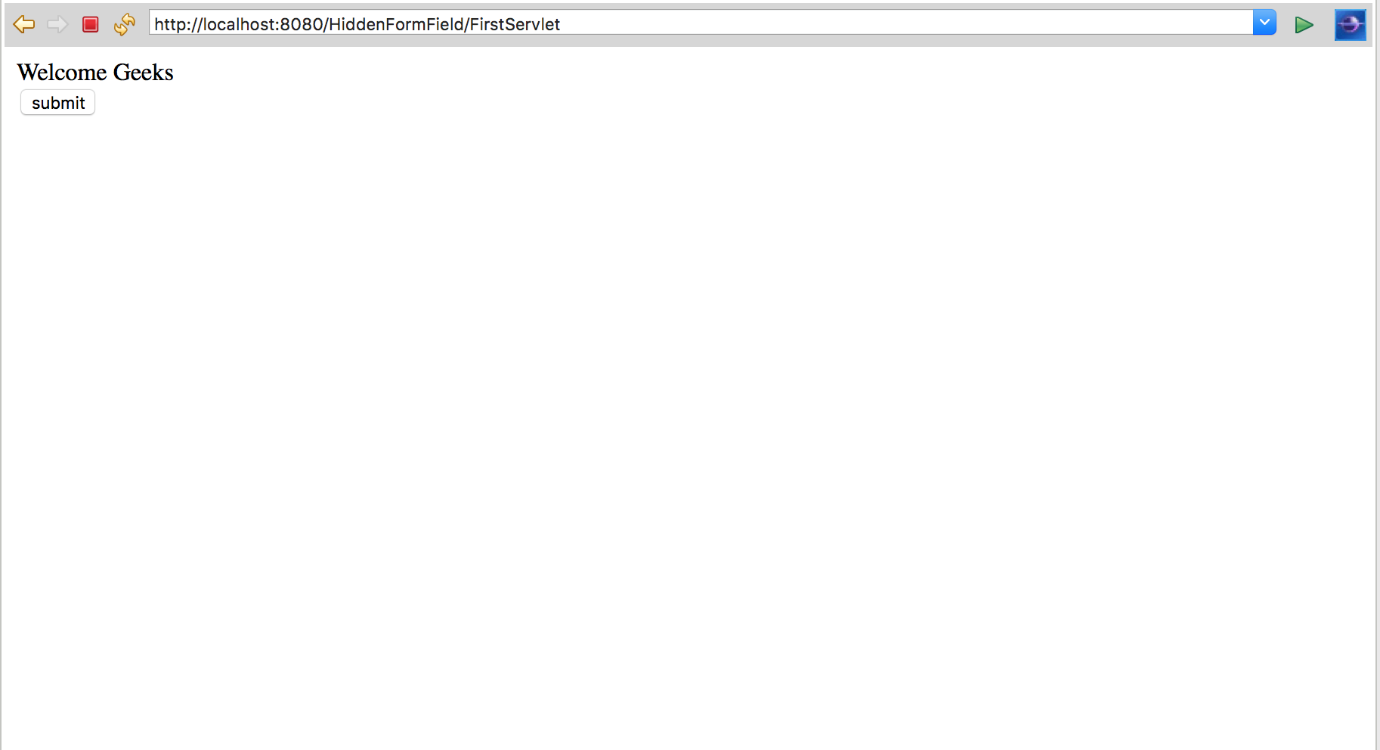
**Output**



**FirstServlet.java**

|  |
| --- |
| // Java program to demonstrate  // Hidden form field method    package GeeksforGeeks;    import java.io.\*;  import javax.servlet.\*;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.\*;    @WebServlet("/FirstServlet")    // this annotation is used for replacing xml file  public class FirstServlet extends HttpServlet {        // class name is FirstServlet which extends HttpServlet      public void doPost(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               \*/              String username = request.getParameter("userName");                /\*               request.getParameter takes the value from index.html file where               name is username               \*/              out.print("Welcome " + username);                // out.println is used to print on the client web browser                /\*               In the below code their is a hidden form  for maintaining session of user.               this passes control to SecondServlet               \*/              out.print("<form action='SecondServlet'>");                out.print("<input type='hidden' name='username' value='" + username + "'>");              out.print("<input type='submit' value='submit'>");              out.print("</form>");              out.close();          }          catch (Exception e) {              System.out.println(e);          }      }  } |

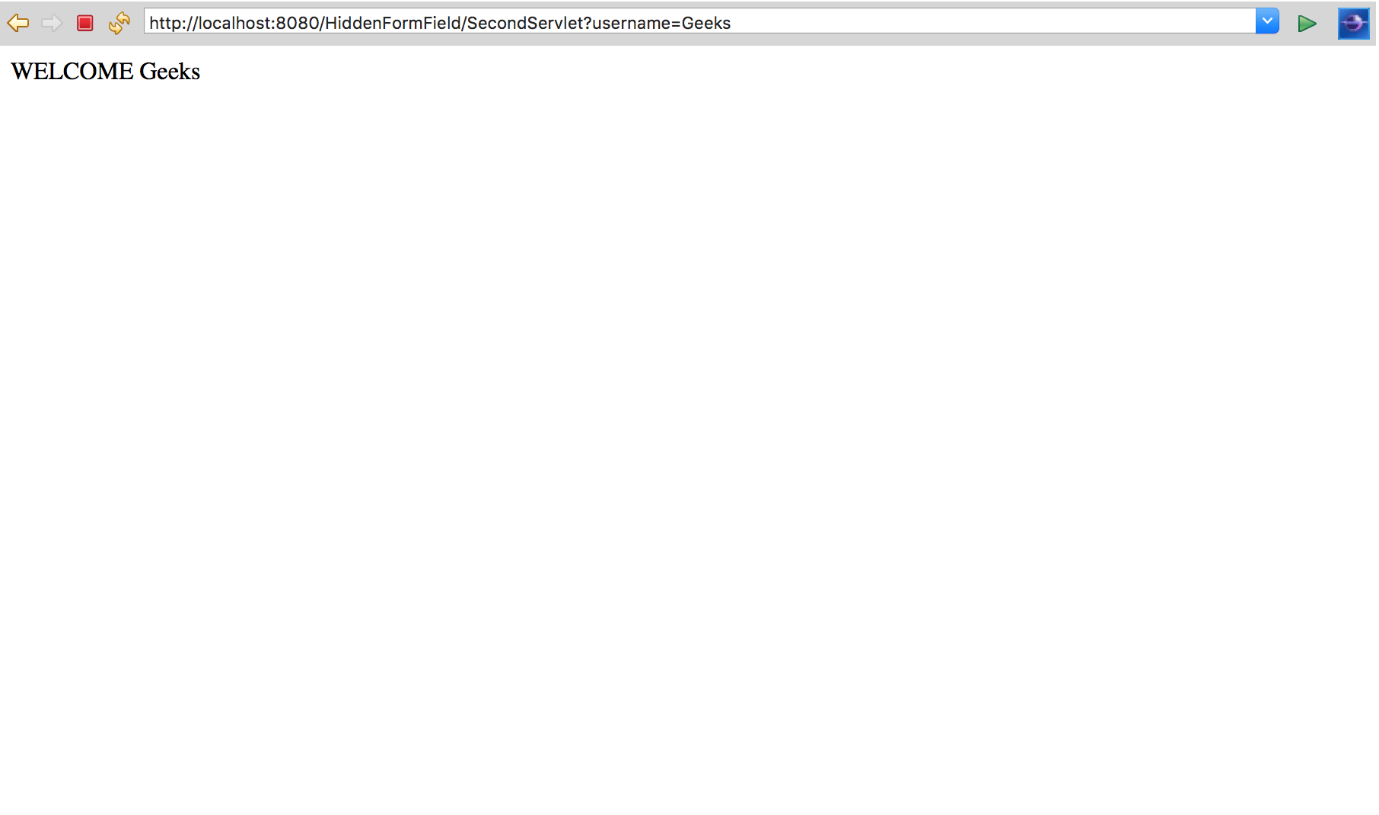
**Output**



**SecondServlet**

|  |
| --- |
| // Java program to demonstrate  // Hidden form field method    package GeeksforGeeks;    import java.io.\*;  import javax.servlet.\*;  import javax.servlet.annotation.WebServlet; // Importing annotation  import javax.servlet.http.\*;    // using this annotation we dont need  // xml file for dispathing servlet  @WebServlet("/SecondServlet")    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 username = request.getParameter("username");                // taking the value of usename from First servlet using getparameter object              out.print("WELCOME " + username);                // out.println is used to print on the client web browser              out.close();          }          catch (Exception e) {              System.out.println(e);          }      }  } |

**Output**



**Step by Step processing of the code**   
As you Deploy the project and run the code on the server following process goes on.

1. Control goes to index.html file and a form is generated on web browser and it will ask for your username.
2. In index.html file form, action is FirstServlet so when you hit submit button, control goes to FirstServlet and your session is start.
3. In FirstServlet request.getParameter(username), take the input from index.html where the field is username and print it on the browser.
4. In firstServlet there is a hidden form field which again takes the input from browser and pass the control to second servlet.
5. This method goes on for a number of different Servlet.   
   Note : This is the main disadvantage of Hidden form field method that you have to fill the form again and again for maintaining a client session.
6. In Second Servlet we again use request.getParameter for getting input from FirstServlet page.

**Advantage and Applications**

* It can be used for anonymous session tracking.
* Hidden form field is supported in every Browser. This method of session tracking does not need any special configuration of the browser.
* All the information is stored in client browser, so it increases the security

**Disadvantage :**

1. It works only for a sequence of dynamically generated forms. This breaks down with static documents, emailed documents, bookmarked documents, and browser shutdowns.
2. You need to submit an extra form on each request.
3. It’s complex than URL rewriting.
4. This method use only textual information.