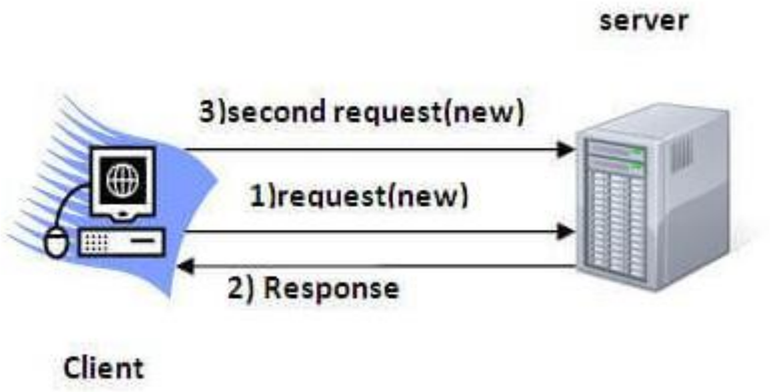
SESSION HANDLING

Session simply means a particular interval of time. **Session Tracking** is a way to maintain state (data) of an user. It is also known as **session management** in servlet.

Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests to the server, server treats the request as the new request. So we need to maintain the state of an user to recognize to particular user.

HTTP is stateless that means each request is considered as the new request. It is shown in the figure given below:



Why use Session Tracking?

To recognize the user It is used to recognize the particular user.

Session Tracking Techniques

There are four techniques used in Session tracking:

1. **Cookies**
2. **Hidden Form Field**
3. **URL Rewriting**
4. **HttpSession**

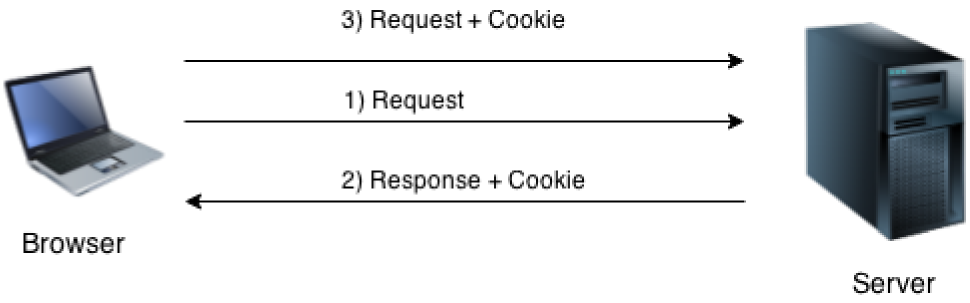
SERVLET COOKIES

A **cookie** is a small piece of information that is persisted between the multiple client requests.

A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

How Cookie works

By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet. So cookie is stored in the cache of the browser. After that if request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user.



Types of Cookie

There are 2 types of cookies in servlets.

1. Non-persistent cookie
2. Persistent cookie

Non-persistent cookie

It is **valid for single session** only. It is removed each time when user closes the browser.

Persistent cookie

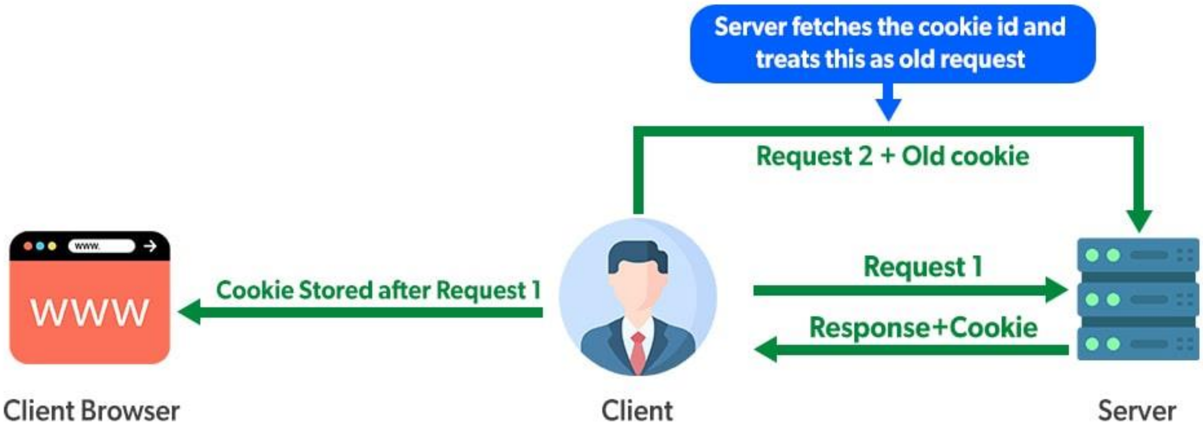
It is **valid for multiple session** . It is not removed each time when user closes the browser. It is removed only if user logout or signout.

Advantage of Cookies

1. Simplest technique of maintaining the state.
2. Cookies are maintained at client side.

Cookies are the textual information that is stored in key-value pair format to the client's browser during multiple requests. It is one of the state management techniques in session tracking. Basically, the server treats every client request as a new one so to avoid this situation cookies are used.

When the client generates a request, the server gives the response with cookies having an id which are then stored in the client's browser. Thus if the client generates a second request, a cookie with the matched id is also sent to the server. The server will fetch the cookie id, if found it will treat it as an old request otherwise the request is considered new.


Server fetches the cookie id and treats this as old request
Request 2 + Old cookie
www.
WWW
Cookie Stored after Request 1
Request 1
Response+Cookie
Client Browser
Client
Server

Cookie class

javax.servlet.http.Cookie class provides the functionality of using cookies. It provides a lot of useful methods for cookies.

Constructor of Cookie class

Constructor	Description
Cookie()	constructs a cookie.
Cookie(String name, String value)	constructs a cookie with a specified name and value.

Useful Methods of Cookie class

There are given some commonly used methods of the Cookie class.

Method	Description
public void setMaxAge(int expiry)	Sets the maximum age of the cookie in seconds.
public String getName()	Returns the name of the cookie. The name cannot be changed after creation.
public String getValue()	Returns the value of the cookie.
public void setName(String name)	changes the name of the cookie.
public void setValue(String value)	changes the value of the cookie.

Other methods required for using Cookies

For adding cookie or getting the value from the cookie, we need some methods provided by other interfaces. They are:

1. **public void addCookie(Cookie ck):**method of HttpServletResponse interface is used to add cookie in response object.
2. **public Cookie[] getCookies():**method of HttpServletRequest interface is used to return all the cookies from the browser.

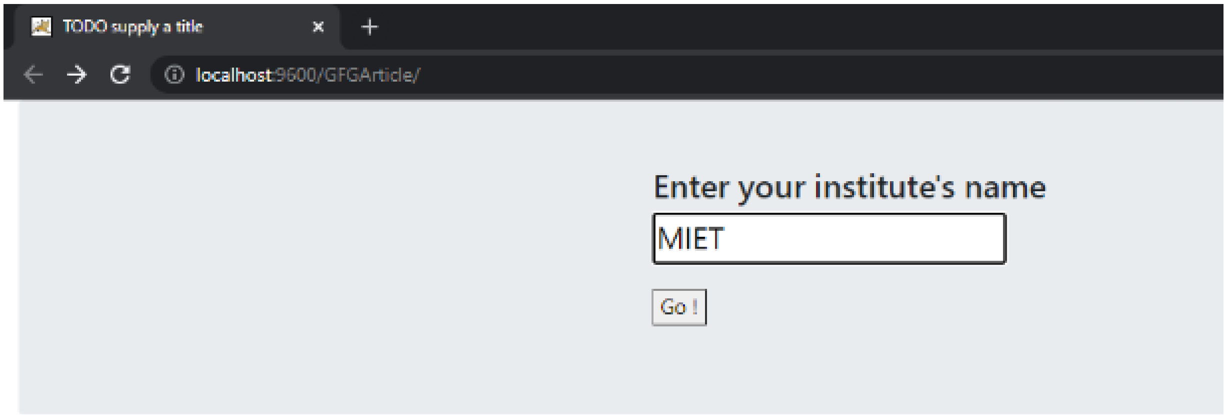
Example

The name of the Institute is passed to Servlet 2 from Servlet 1 using Cookies.

```
<!DOCTYPE html>

<html>

        <head>

                <title>TODO supply a title</title>

                <!-- css-->

                <link rel="stylesheet"
href=https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css
        integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"

                                                crossorigin="anonymous">

                <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

                <meta charset="UTF-8">

                <meta name="viewport" content="width=device-width, initial-scale=1.0">

        </head>
```

```
    <body>

        <form action="servlet1" method="POST">

            <div class="container-fluid ">

                <div class="jumbotron">

                <div class="container col-sm-4">

            <h2>Enter your institute's name</h2>

            <input type="text" name="name" style="font-size:30px;">

            <br>

            <br>

            <!-- button to redirect to servlet1 -->

            <button type="submit" style="font-size:20px;" class="center">

                Go !

            </button>

            <br><br>

            </div>

            </div>

            </div>

        </form>

    </body>

</html>
```

Output:



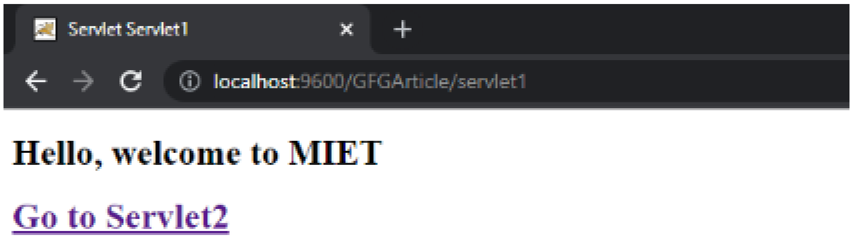
Enter your institute's name

MIET

```
import jakarta.servlet.*;

import jakarta.servlet.http.*;

import java.io.IOException;

import java.io.PrintWriter;

public class Servlet1 extends HttpServlet {

        protected void

        processRequest(HttpServletRequest request,

                                HttpServletResponse response)

                throws ServletException, IOException

        {

                response.setContentType("text/html;charset=UTF-8");

                try (PrintWriter out = response.getWriter()) {

                        out.println("<!DOCTYPE html>");

                        out.println("<html>");

                        out.println("<head>");

                        out.println("<title>Servlet Servlet1</title>");

                        out.println("</head>");

                        out.println("<body>");
```

```
                    // Creating a string to store the name
                    String name = request.getParameter("name");
                    out.println("<h1> Hello, welcome to " + name + " </h1>");
                    out.println( "<h1><a href =\"servlet2\">Go to Servlet2</a></h1>");
                    // Creating a cookie
                    Cookie c = new Cookie("user_name", name);
                    response.addCookie(c);
                    out.println("</body>");
                    out.println("</html>");
                }
        }
}
```

Output

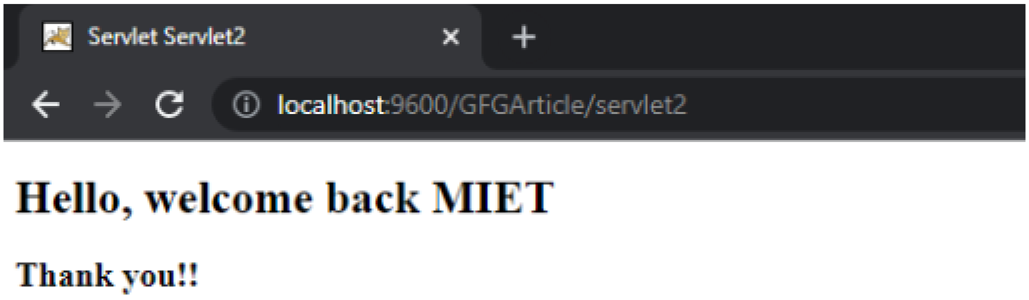


Hello, welcome to MIET

Go to Servlet2

```
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;

public class Servlet2 extends HttpServlet {

        protected void
        processRequest(HttpServletRequest request,
                HttpServletResponse response)
            throws ServletException, IOException
        {
            response.setContentType("text/html;charset=UTF-8");
            try (PrintWriter out = response.getWriter()) {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>Servlet Servlet2</title>");
                out.println("</head>");
                out.println("<body>");

                // Fetching cookies(if found more than one)
                // Array of Cookies
                Cookie[] cookies = request.getCookies();
                boolean f = false;
                String name = "";
                if (cookies == null) {
                    out.println("<h1>You are new user, go to home page and submit your institute's name");
                    return;
                }
                else {
                    for (Cookie c : cookies) {
                        String tname = c.getName();
                        if (tname.equals("user_name")) {
                            f = true;
                            name = c.getValue();
                        }
                    }
                }
                if (f) {
                    out.println("<h1> Hello, welcome back " + name + " </h1>");
                    out.println("<h2>Thank you!!</h2>");
                }
                else {
                    out.println(
                        "<h1>You are new user, go to home page and submit your institute's name");
                }

                out.println("</body>");
                out.println("</html>");
            }
        }
    }
```

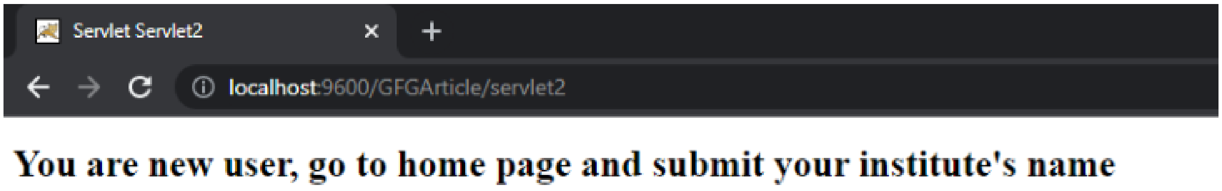
Output



Hello, welcome back MIET

Thank you!!

If one runs the servlet2 link in incognito mode directly, cookies are not recognized and it is treated as a new user



You are new user, go to home page and submit your institute's name

Disadvantages of cookies

- Space character and image are considered invalid.
- Security is less.