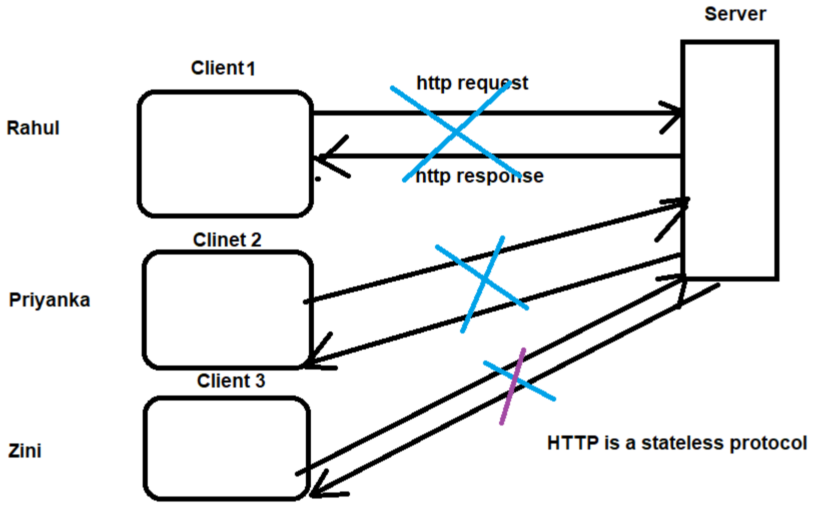
DAY: 30, Date: 20/07/2024
Chapter: 7, Name: State Management

- Normally whenever we are sending any request to server those requests are HTTP type.
- HTTP is a stateless protocol.
- 

- HTTP is a stateless protocol. It means once the response is delivered by the server then entire page information will be destroyed. This is the drawback of HTTP protocol.
- Stateless protocol means unable to remember the client information.
- If the server is unable to remember the client information, then that server will treat each request as a new request.
- To overcome the drawback of stateless mechanisms we must use state management techniques.
- State management is a process to maintain the state of the client or webpage.
- Here the meaning of the state is username, password, email address, etc.
- State value is maintained by either client/server.
- Some of the popular state management techniques are :
    - **COOKIES**
    - **SESSION**
    - **URL rewriting**
    - **Hidden Form Field**
    - **Cache etc.**
- State management is also called as Session management.

Chapter: 7.1(COOKIES)

- It is a state management technique.
- Cookies are a small amount of information.
- Server will create the cookie and client will maintain cookie.
- Maintain means store.
- Cookies can store 4KB of information which is equivalent to 4096 bytes.
- In the cookie data stores in the form of plain text.
- For a website, a browser can allow maximum 20 cookies.
- If it cross more than 20 cookies, then automatically it will deleted the old one.(21 –> ~~1~~, 22- >~~2~~ , 23->~~3~~).

Types of Cookies

- In-memory cookies
- Persistent cookies

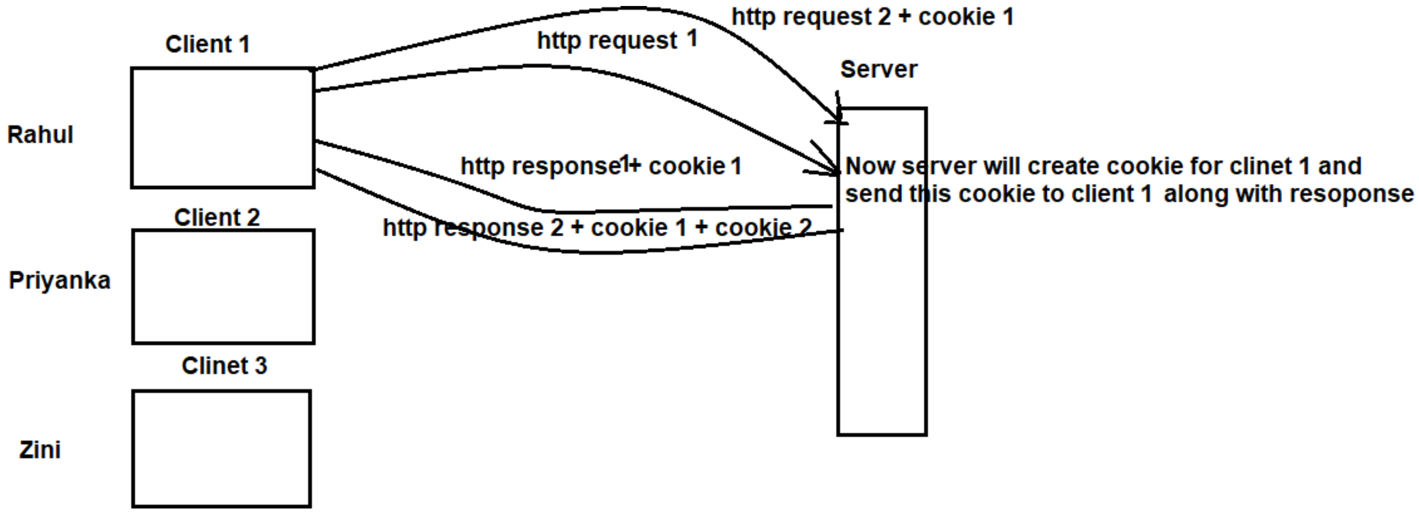
<u>In-memory cookies:</u>

- In in-memory cookies the cookie will be stored in browser memory, when the browser will close then cookie will be destroyed.

Persistent cookies:

- Persistent cookies have expiry date and time.
- If the browser close, then still it hold cookie value inside the browser memory.

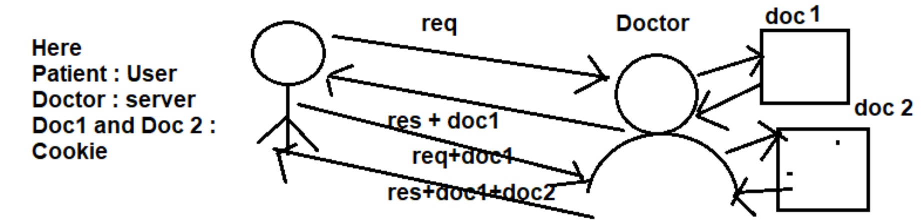
Diagram of client -server communication with cookie



Now server will create cookie for clinet 1 and send this cookie to client 1 along with resoponse

Client server communication with cookies

Here in this diagram client 1 sends http request 1 to server. Now the server receives the request from client 1. If the server wants to remember the client information, then the server will create a cookie object then server will send the cookie object along with the response to client 1. When client 1 sends more requests to the server, server will not treat as a new request.



Here doctor(server) created the cookie and patient(client) maintain the cookie.

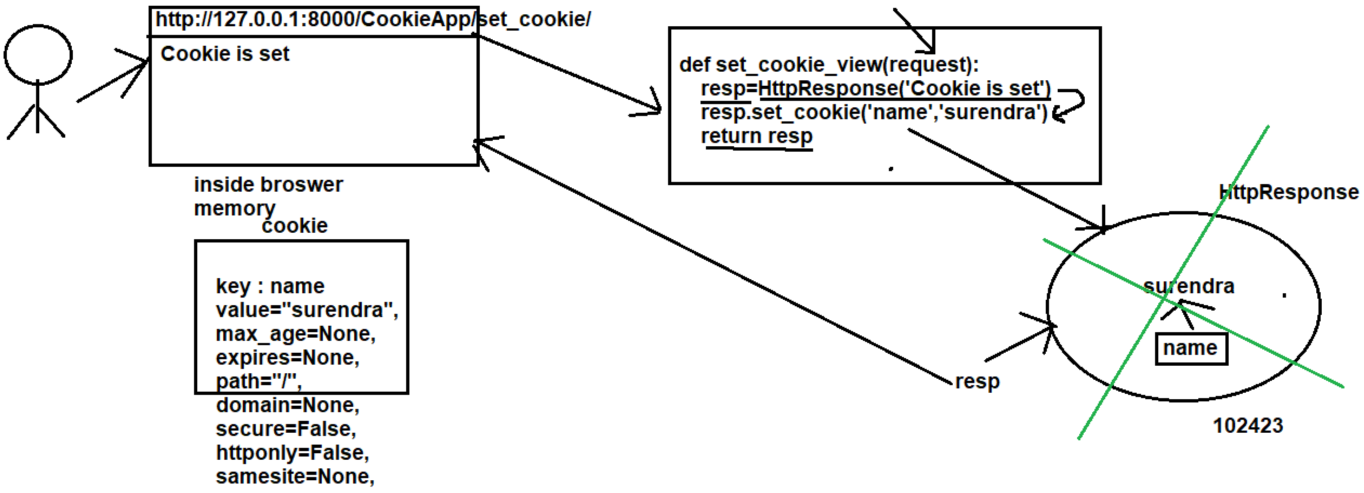
Real life example

Day: 31 and Date: 21/07/2024
Topic: Implementation of Cookies

a) How to create the cookie?

b) How to get the cookie?
c) How to update the cookie?
d) How to delete the cookie?

a) How to create the cookie:



```
def set_cookie_view(request):
   resp=HttpResponse('Cookie is set')
   resp.set_cookie('name','surendra')
   return resp
```

```
key : name
value="surendra",
max_age=None,
expires=None,
path="/",
domain=None,
secure=False,
httponly=False,
samesite=None,
```

b) How to set,get and delete the cookie

```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def set_cookie_view(request):
    resp=HttpResponse('Cookie is set')
    resp.set_cookie('name','surendra')
    return resp

def get_cookie_view(request):
    name=request.COOKIES['name']
    return HttpResponse('Cookie Name is '+name)

def delete_cookie_view(request):
    resp=HttpResponse('Cookie deleted')
    resp.delete_cookie('name')
    return resp
```