

SESSION:

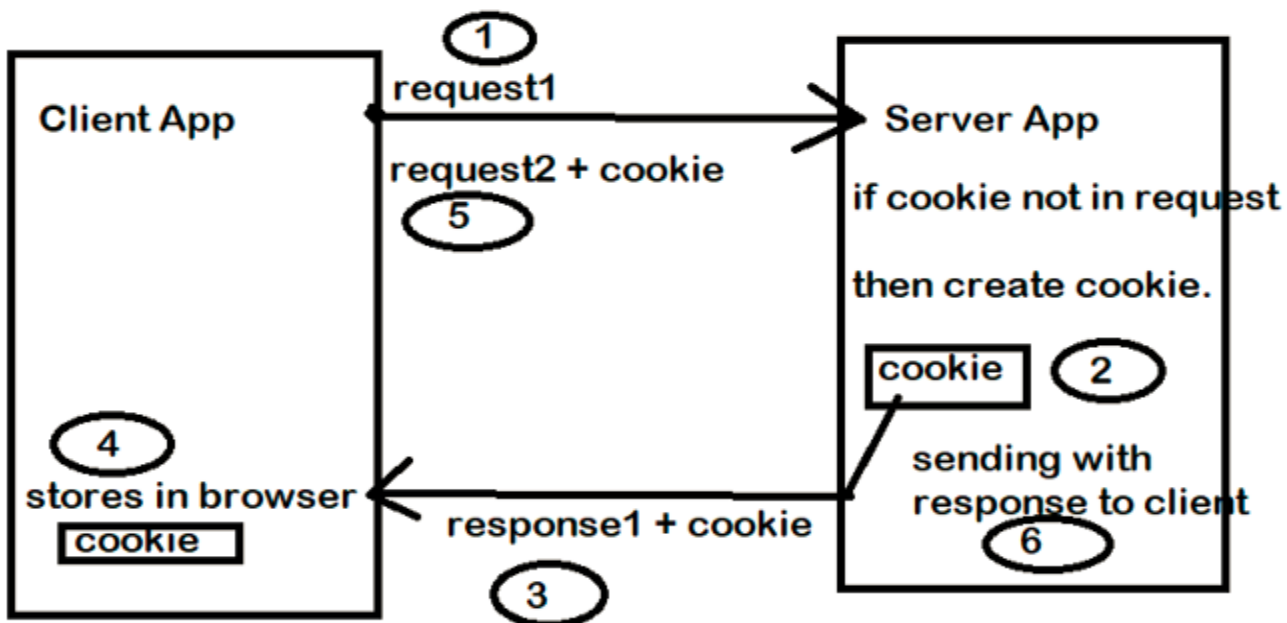
- The set of continuous requests which are performed by one client on one web application is known as "session".
- For example : Login to Logout
- During the session, by default previous request data can not be remembered in the current request. Because we are using "http" protocol.
- http protocol is a "stateless" protocol
- Because of the stateless protocol, our web application behaves like as a stateless application
- By implementing session tracking techniques in our application we can provide "statefull" behaviour to the web application.

COOKIES:

- Cookies are small textual information, which are created at server side application and stored in clients computer
- Cookies transferred from server to client and client to server through the response and request objects

How Cookie will work :

1. The browser sends the request to the server.
2. The server sends the Response along with one or more cookie to the browser.
3. The browser saves the cookie which sent from the server.
4. From now on words the browser will send this cookie to the server, every time a request is made to the server.
5. The browser will keep sending the cookie to the server along with each request untill the cookie expires.
6. When the cookie expires, it will removed from the browser.



Client and Server Apps Architecture

Creating Cookies in Django

The `set_cookie()` has these attributes:

name: It specifies the name of cookie.

value: It specifies the text or variable you want to store in the cookie.

max_age: It is the time period of cookie in seconds.

- After the time period completes, it will expire.
- It is an optional parameter; if not present then the cookie will exist till the time browser close.

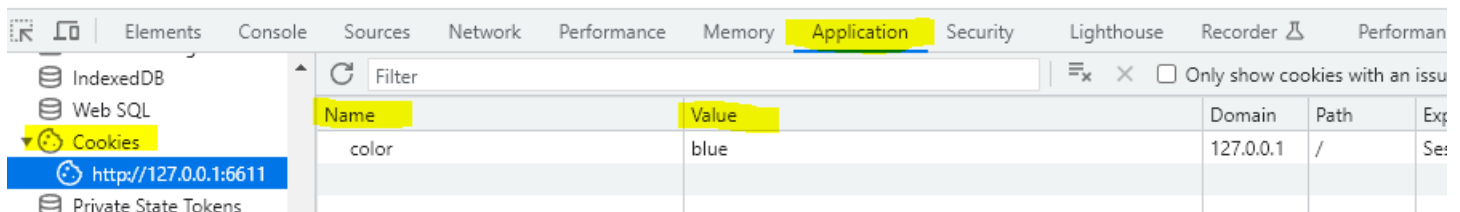
Syntax: `set_cookie("Cookie_Name", "Cookie_Value", max_age = None)`

We will add this `set_cookie()` to our view functions for creating cookies.

For example:

```
def setcookie(request):  
    response = HttpResponse("<h1>Cookie created now. </h1>")  
    resp.set_cookie('data', 'Hello this is your Cookies', max_age = None)  
    return response
```

- This function will return response to the browser, and when you run it, you can check the created cookie in your browser settings.
- To see cookie in browser , press F12 in browser, Click on Application, Click on Cookies , Click on URL.



We have two types of cookies in django. They are,

1. Inmemory cookies
2. Persistence cookies

1. Inmemory cookies:

- The cookies which doesn't contain explicit expire time are known as "inmemory" cookies
- Inmemory cookies will be stored in the clients machine primary memory.
- When ever browser is closed, automatically inmemory cookies will be deleted.

2. Persistence cookies:

- The cookies which contains explicit expire time are called "persistence cookies"
- These will be stored in the client machine hard disk. So , these will be deleted when ever expire time will be overed
- Even though we close the browser window, persistence cookies will not be deleted with out expire time is overed
- We can create and add the cookies to the response object by calling `set_cookie()` method of `HttpResponse`
- We can get the cookies from the Request object by using `COOKIES` attribute.
- We can create the persistence cookies in django application as follows

```
response = HttpResponse()  
response.set_cookie('cookie_name', 'cookie_value', max_age=60)
```