**SESSOION MANAGEMENT IN DJANGO**

HTTP is a stateless protocol, which means the server does not maintain a continuous connection. Once the server handles the request and sends back the response that is there is no socket connection alive with the socket connection is destroyed.

When the client send another request, the client has to establish another new connection and server will handle it as a fresh request and then it sends the response back.

Advantages:

Performance

Scalability

Disadvantages:

Session tracking

**Session Tracking**

Maintaining the state across the application request despite the stateless of the history of http protocol that is if we want to maintain something, then we have to use Session Tracking.

**Session:**

Two ways:

1. From the user login in until he logout, can be read as a session. E.g.: Facebook / Gmail.
2. Tracking user interaction : From the time we access the website for the first time till you close the browser is called a session. E.g.: eCommerce, ticket booking.

First way of using a session tracking is using cookies, which are a http feature so when the client request comes into our application, the application will process a send response and in the response it will set a cookie.

Second ways is Session API support the Django has. When the user request comes in, we use the Django Session API by simply using request.session the Django server will internally create a session for us and could generate a unique id. Every session on this Django server / client session will be assigned a unique id and that id will be sent back to the client as cookie and that session ID can be mapped to a particular dictionary and we can store our details inside this dictionary.

**Cookie Support**

request.session

set\_test\_cookie() : Will set a random cookie method for the web browser.

test\_cookie\_worked()

delete\_test\_cookie()

Step 1: Create a project and an app.

Step 2: Inside project.settings.py add the app name

Step 3: In views.py add the below:

views.py

from django.shortcuts import render  
from django.http import HttpResponse  
  
  
*# Create your views here.*def home(request):  
 request.session.set\_test\_cookie()  
 return HttpResponse("<h1> Home Page </h1>")  
  
def main(request):  
 if request.session.test\_cookie\_worked():  
 print("Cookies are enabled !!!!")  
 request.session.delete\_test\_cookie()  
 print("Cookies Deleted !!!!")  
 return HttpResponse("<h1> Main Page </h1>")

Step 4: In urls.py file of project, import the views and add the line to map the project

from cookieApp import views  
  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('home/', views.home),  
 path('main/', views.main)  
]

Step 5: Execute these below:

PS C:\Users\kamal\django\_projects\cookiesProject> py -m manage makemigrations

No changes detected

PS C:\Users\kamal\django\_projects\cookiesProject> py -m manage migrate

Step 6: For testing run the cmd below:

PS C:\Users\kamal\django\_projects\cookiesProject> py -m manage runserver

**Output**

A screenshot of a computer

Description automatically generated

A white rectangular object with a black border

Description automatically generated with medium confidence

**Output in Console**

[17/May/2024 22:35:47] "GET /home/ HTTP/1.1" 200 20

Cookies are enabled !!!!

Cookies Deleted !!!!

[17/May/2024 22:35:53] "GET /main/ HTTP/1.1" 200 20