Python | Sessions framework using django

Last Updated : 07 Nov, 2023

The sessions framework can be used to provide persistent behavior for anonymous users on the website. Sessions are the mechanism used by Django for you to store and retrieve data on a per-site-visitor basis. Django uses a cookie containing a unique session ID.

Django Sessions

So let's understand what a is session in Django and how to use sessions. Django provides full support for anonymous sessions. The session framework lets you store and retrieve arbitrary data on a per-site-visitor basis. It keeps data on the server side and abstracts the sending and receiving of cookies. Now that we have some basic understanding of the session let's figure out how to use session in Django.

How to use session in Django

First, enable Session in Django. To enable the session in Django, you will need to make sure of two things in `settings.py`:

MIDDLEWARE_CLASSES has 'django.contrib.sessions.middleware.SessionMiddleware' activate
INSTALLED_APPS has 'django.contrib.sessions' added.

Python3

```
# Application definition
INSTALLED APPS = [
'dhun',
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
]
MIDDLEWARE = [
'django.middleware.securitY.SecuritYMiddleware',
'django.contrib.sessions.middleware.SessionMiddLeware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMidd1eware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

After enabling the session, the session database table has to create and to do this run the following command:

Also, ensure that the django.contrib.sessions app is included in the INSTALLED_APPS list:

Python3

```
INSTALLED_APPS = [
    # ...
    'django.contrib.sessions',
    # ...
]
```

```
python manage.py syncdb
```

After running previous command and if it didn't find any errors then later run the command given below to finally reflect the changes saved onto the migration file onto the database.

```
python manage.py migrate
```

Now once sessions are created, then testing of the cookies has to be done.

Configure Session Storage

By default, Django uses a database-backed session storage. You can configure it in your settings file. For example, to use the database-backed session storage

```
SESSION_ENGINE = 'django.contrib.sessions.backends.db'
```

If you prefer using cache-based sessions (which is often faster), you can configure it like this:

```
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
```

Additionally, configure the cache settings in your settings file:

Python3

```python
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
    }
}
```

Using Sessions

In `views.py`, set the test cookie in the index view, and test the cookie in your about view.

Python3

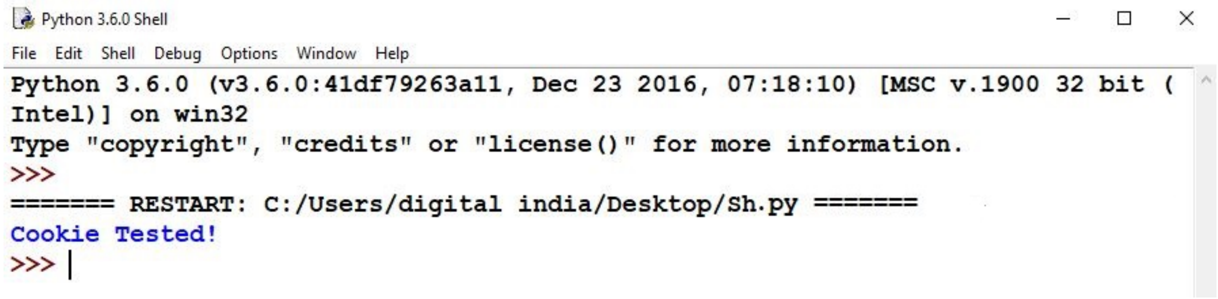
```python
from django.shortcuts import render
from django.http import HttpResponse
from .models import Album

def home(request) :
    a = Album.objects.all()
    return render(request, "dhun/home.html ", {"Album":a})

def index(request) :
    num_authors = Author.objects.count()
    request.session.set_test_cookie()
    num_visits = request.session.get( 'num_visits', 0)
    request.session ['num_visits'] = num_visits + 1
    context ={
        'num_books':num_books,
        'num_instances':num_instances,
        'num_instances available':num_instances_available,
        'num_authors':num_authors,
        'num_visits':num_visits
```

```
def about(request):
    LANGUAGE_CODE ='en-us '
    TIME_ZONE ='UTC'
    if request.session.test_cookie_worked():
        print ("Cookie Tested !")
        request.session.delete_test_cookie()
```

Now, First run the localhost through this command.

```
python manage.py runserver
```

Then Open http://localhost:8000/ in the browser. Visit the index page then visit the about page. The “Cookie Tested!” will be printed out to the console.



```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/digital india/Desktop/Sh.py =======
Cookie Tested!
>>> |
```

To know how many times the site has been visited.You have to do following two things In views.py:

- Add and update the code in the index view function/li>
- Update the about view function

Python3

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

```
def home(request):
    a = AIbum. objects.all()
    return render(request, "dhun/home.html", {"album":a})

def index(request):
    visits = int(reques.COOKIES.get('visits', '0'))
    response = HttpResponse(template.render(context))

    if request.COOKIES.has_key('last_visit'):
        last_visit = request. COOKIES [ ' last_visit']
        last_visit_time = datetime.strptime(last_visit[:-7], "%Y-%m-%d %H:%M:%S")
        curr_time = datetime.now()
        if (curr_time—last_visit_time).days > O:
            response.set_cookie( 'visits ', visits + 1)
            response. set_cookie( ' last_visit', datetime.now())
        else :
            response.set_cookie( ' last_visit', datetime.now())
        return response

def about(request) :
    context = RequestContext(request)
    if request.COOKIES.has_key(' visits '):
        v = request.COOKIES [' visits ']
    else :
        v = 0
    return render_to_response('music/about.html', { 'visits':v}, context)
```

Session Expiry

By default, Django sessions expire when the user’s browser is closed. You can configure session expiry by setting the SESSION_COOKIE_AGE setting in your settings.py. For example, to set a session timeout of 30 minutes:

```
SESSION_COOKIE_AGE = 1800  # 30 minutes (in seconds)
```

That’s it! You can now use the Django sessions framework to manage user-specific data across your web application. Remember to handle session data carefully, especially if it’s sensitive information or related to user authentication.

S SHUB... 4

Previous Article

Django - Dealing with warnings

Next Article

Django Sign Up and login with confirmation Email | Python

Similar Reads

Integrating Django with Reactjs using Django REST Framework

In this article, we will learn the process of communicating between the Django Backend and React js frontend using the Django REST Framework. For the sake...

15+ min read

Implement Token Authentication using Django REST Framework

Token authentication refers to exchanging username and password for a token that will be used in all subsequent requests so to identify the user on the server...

2 min read

How to Create a basic API using Django Rest Framework ?

Django REST Framework is a wrapper over the default Django Framework, basically used to create APIs of various kinds. There are three stages before...

4 min read

Creating and Using Serializers - Django REST Framework

In Django REST Framework the very concept of Serializing is to convert DB data to a datatype that can be used by javascript. Serializers allow complex data suc...

3 min read

Rendering Data-Frame to html template in table view using Django…

In Django, It is easy to render the HTML templates by setting URLs of respective HTML pages. Here we will let to know about how we can work with DataFrame...

3 min read

Joke Application Project Using Django Framework

Django is a high-level Python-based Web Framework that allows rapid development and clean, pragmatic design. It is also called batteries included...

2 min read

Create a Counter App Using React, Tailwind and Django Framework

This article will guide you in creating a Counter using React and Tailwind with the Django Framework. We'll explore the integration of Django, React, and Tailwind...

6 min read

OTP Verification in Django REST Framework using JWT and Cryptography

In this article, we'll explore how to implement OTP verification in Django REST Framework using JWT and cryptography. We will guide you through the proces...

6 min read

Voting System Project Using Django Framework

Project Title: Pollster (Voting System) web application using Django frameworkType of Application (Category): Web application. Introduction: We wil...

14 min read

Django REST Framework Installation

Django REST Framework can be installed via pip package similar to Django installation. Since Django REST Framework is a wrapper over default Django...

1 min read

Article Tags : Python Python Django

Practice Tags : python

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company
About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects