



Django Request and Response cycle – HttpRequest and HttpResponse Objects

Last Updated : 27 May, 2021

Prerequisite – [Views In Django | Python](#)

Before we jump into using convenience methods that views come with, Let's talk about the Request and Response cycle. So, when a Request happens to a Django server, a couple of things go on. One of them is Middleware.

Middleware

Middleware is like a middle ground between a request and response. It is like a window through which data passes. As in a window, light passes in and out of the house. Similarly, when a request is made it moves through middlewares to views, and data is passed through middleware as a response.

Here are the default middlewares installed in Django.

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

You can add your middlewares. We will be discussing that in the next articles.

Request and Response Objects:

Django uses request and response objects to pass state through the system.

metadata about the request. Then Django loads the appropriate view, passing the HttpRequest as the first argument to the view function. Each view is responsible for returning an HttpResponse object.

HttpRequest and HttpResponse object Example

- To explain these objects let's create a view home as below in **views.py**

Python3

```
# importing HttpResponse from library
from django.http import HttpResponse

def home(request):
    # request is handled using HttpResponse object
    return HttpResponse("Any kind of HTML Here")
```

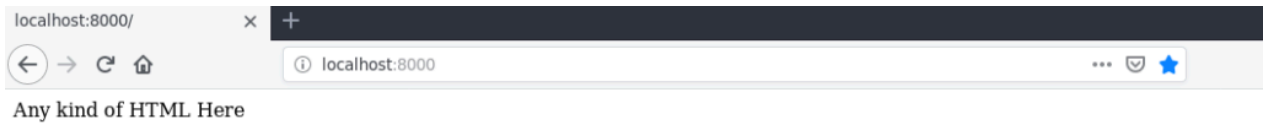
- To handle the request let us map a URL to this view in **urls.py**

Python3

```
# importing view from views.py
from .views import home

urlpatterns = [
    path('', home),
]
```

- Now you can run the server to see the following in the browser



HttpRequest Attributes – Django

You can use the following attributes with HttpRequest for advanced manipulation

Attribute	Description
HttpRequest.scheme	A string representing the scheme of the request (HTTP or HTTPS usually).
HttpRequest.body	It returns the raw HTTP request body as a byte string.
HttpRequest.path	It returns the full path to the requested page does not include the scheme or domain.
HttpRequest.path_info	It shows path info portion of the path.
HttpRequest.method	It shows the HTTP method used in the request.
HttpRequest.encoding	It shows the current encoding used to decode form submission data.
HttpRequest.content_type	It shows the MIME type of the request, parsed from the CONTENT_TYPE header.

Attribute	Description
HttpRequest.content_params	It returns a dictionary of key/value parameters included in the CONTENT_TYPE header.
HttpRequest.GET	It returns a dictionary-like object containing all given HTTP GET parameters.
HttpRequest.POST	It is a dictionary-like object containing all given HTTP POST parameters.
HttpRequest.COOKIES	It returns all cookies available.
HttpRequest.FILES	It contains all uploaded files.
HttpRequest.META	It shows all available Http headers.
HttpRequest.resolver_match	It contains an instance of ResolverMatch representing the resolved URL.

HttpRequest Methods – Django

You can use the following methods with HttpRequest for advanced manipulation

Attribute	Description
HttpRequest.get_host()	It returns the original host of the request.
HttpRequest.get_port()	It returns the originating port of the request.
HttpRequest.get_full_path()	It returns the path, plus an appended query string, if applicable.

Attribute	Description
<code>HttpRequest.build_absolute_uri(location)</code>	It returns the absolute URI form of location.
<code>HttpRequest.get_signed_cookie(key, default=RAISE_ERROR, salt="", max_age=None)</code>	It returns a cookie value for a signed cookie, or raises a <code>django.core.signing.BadSignature</code> exception if the signature is no longer valid.
<code>HttpRequest.is_secure()</code>	It returns True if the request is secure; that is, if it was made with HTTPS.
<code>HttpRequest.is_ajax()</code>	It returns True if the request was made via an XMLHttpRequest.

HttpResponse Attributes – Django

You can use the following attributes with HttpResponse for advanced manipulation

Attribute	Description
<code>HttpResponse.content</code>	A bytestring representing the content, encoded from a string if necessary.
<code>HttpResponse.charset</code>	It is a string denoting the charset in which the response will be encoded.
<code>HttpResponse.status_code</code>	It is an HTTP status code for the response.
<code>HttpResponse.reason_phrase</code>	The HTTP reason phrase for the response.
<code>HttpResponse.streaming</code>	It is false by default.

Attribute	Description
HttpResponse.closed	It is True if the response has been closed.

HttpResponse Methods – Django

You can use the following methods with HttpResponse for advanced manipulation

Method	Description
<code>HttpResponse.__init__(content="", content_type=None, status=200, reason=None, charset=None)</code>	It is used to instantiate an HttpResponse object with the given page content and content type.
<code>HttpResponse.__setitem__(header, value)</code>	It is used to set the given header name to the given value.
<code>HttpResponse.__delitem__(header)</code>	It deletes the header with the given name.
<code>HttpResponse.__getitem__(header)</code>	It returns the value for the given header name.
<code>HttpResponse.has_header(header)</code>	It returns either True or False based on a case-insensitive check for a header with the provided name.
<code>HttpResponse.setdefault(header, value)</code>	It is used to set default header.
<code>HttpResponse.write(content)</code>	It is used to create response object of file-like object.

Method	Description
HttpResponse.flush()	It is used to flush the response object.
HttpResponse.tell()	This method makes an HttpResponse instance a file-like object.
HttpResponse.getvalue()	It is used to get the value of HttpResponse.content.
HttpResponse.readable()	This method is used to create stream-like object of HttpResponse class.
HttpResponse.seekable()	It is used to make response object seekable.

Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

N Nave...



6

Asynchronous HTTP Requests with Python

Similar Reads

response.request - Python requests

response.request returns the request object that requested this response. Python requests are generally used to fetch the content from a particular resource URI....

2 min read

Python Falcon - Request & Response

Python Falcon, a nimble web framework, simplifies the process of building web APIs by efficiently managing incoming requests, which are messages asking for...

7 min read

Creating a JSON Response Using Django and Python

In Django, we can give responses to the front end in several ways. The simplest way to render a template or send a JSON response. JSON stands for JavaScript...

4 min read

Render a HTML Template as Response - Django Views

A view function, or view for short, is simply a Python function that takes a Web request and returns a Web response. This article revolves around how to render...

3 min read

How Do we Raise a Response Forbidden in Django?

In Django, this "Response Forbidden" should be risen in a case when the user is not authorized or have permission to get some resource or perform some action....

5 min read

Handling Ajax request in Django

Introduction This tutorial explains how to carry out an ajax request in the Django web framework. We will create a simple post-liking app as a part of the exampl...

4 min read

How to get GET request values in Django?

Django, a high-level Python web framework, simplifies the process of web development. One common task in web applications is handling GET requests,...

2 min read

Get Request.User in Django-Rest-Framework Serializer

In Django Rest Framework (DRF), serializers are used to transform complex data types, such as queryset and model instances, into native Python data types. One...

5 min read

How to get JSON data from request in Django?

Handling incoming JSON data in Django is a common task when building web applications. Whether you're developing an API or a web service, it's crucial to...

2 min read

cycle - Django Template Tags

A Django template is a text document or a Python string marked-up using the Django template language. Django being a powerful Batteries included...

3 min read

Article Tags : [Python](#) [Django-basics](#)

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

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

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

Languages

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

Data Science & ML

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

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

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

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

Interview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved