



[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

Django Introduction | Set 2 (Creating a Project)

Last Updated : 10 Dec, 2020

Note- This article is in continuation of [Django introduction](#).

Popularity of Django

Django is used in many popular sites like as: Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic etc. There are more than 5k online sites based on Django framework. ([Source](#))

Sites like [Hot Frameworks](#) assess the popularity of a framework by counting the number of GitHub projects and StackOverflow questions for each platform, here Django is in 6th position. Web frameworks often refer to themselves as “opinionated” or “un-opinionated” based on opinions about the right way to handle any particular task. Django is somewhat opinionated, hence delivers the in both worlds(opinionated & un-opinionated).

Django architecture

Django is based on MVT(Model-View-Template) architecture. MVT is a software design pattern for developing web application. MVT have following three parts-

- **Model-** Model is going to act as interface of your data. It is responsible for maintaining data.
- **View-** View is gonna be the interface that user see.
- **Template** – A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

Project Structure- Inside the `geeks_site` folder (project folder) there will be following files-

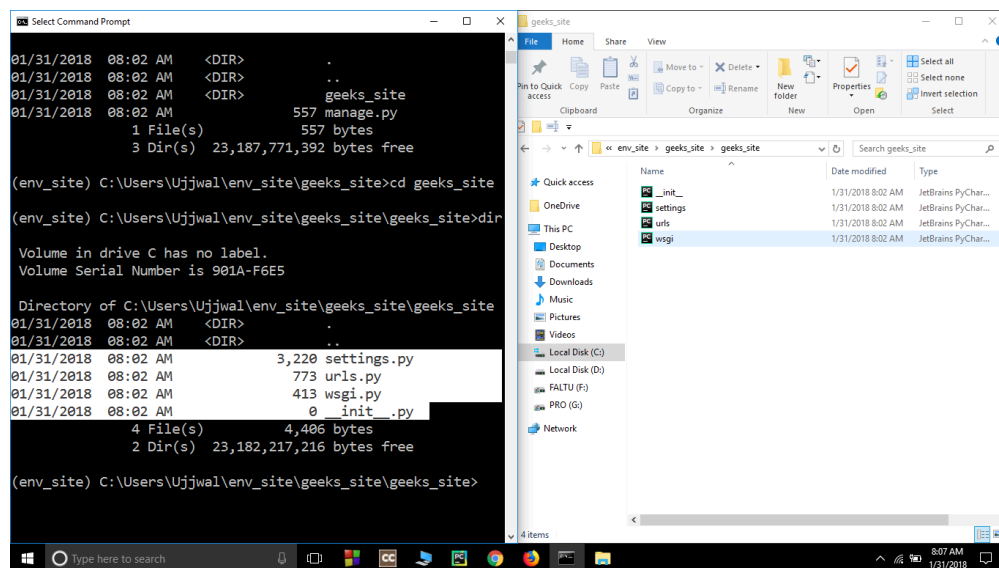
manage.py- This file is used to interact with your project via command line(start the server, sync the database... etc). For getting the full list of

command that can be executed by manage.py type this code in command window –

```
python manage.py help
```

folder (`geeks_site`) –This folder contains all the packages of your project. Initially, it contains four files –

- **`_init_.py`** – It is a python package.
- **`settings.py`** – As the name indicates it contains all the website settings. In this file we register any applications we create, the location of our static files, database configuration details, etc.
- **`urls.py`** – In this file we store all links of the project and functions to call.
- **`wsgi.py`** – This file is used in deploying the project in WSGI. It is used to help your Django application communicate with the web server.



Create a project

Assuming you have gone through [this](#) article and have setup django successfully.

Create a new file **`views.py`** inside the project folder where `settings.py`, `urls.py` and other files are stored and save the following code in it-

Python3

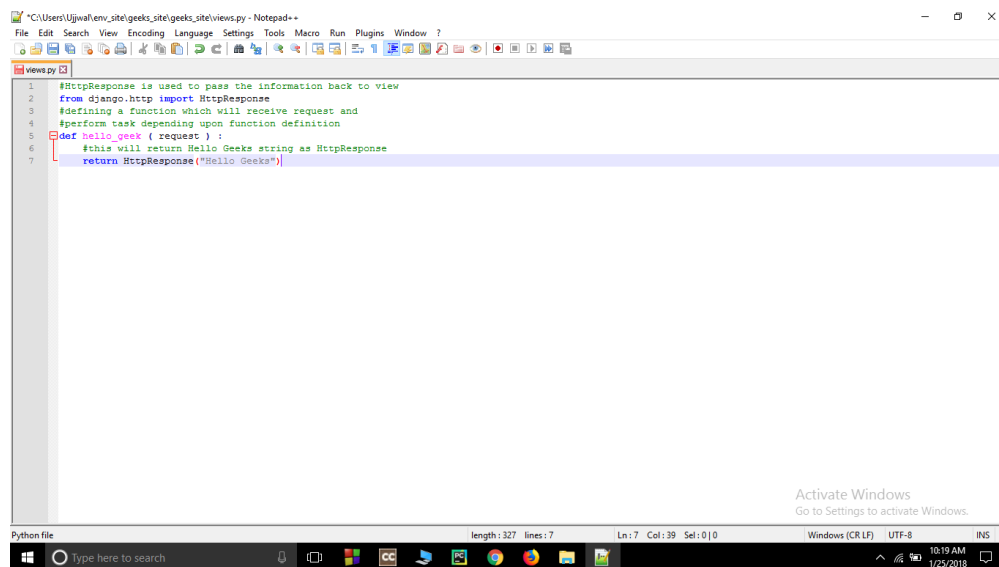
```
# HttpResponse is used to
```

```
# pass the information
# back to view
from django.http import HttpResponse

# Defining a function which
# will receive request and
# perform task depending
# upon function definition
def hello_geek (request) :

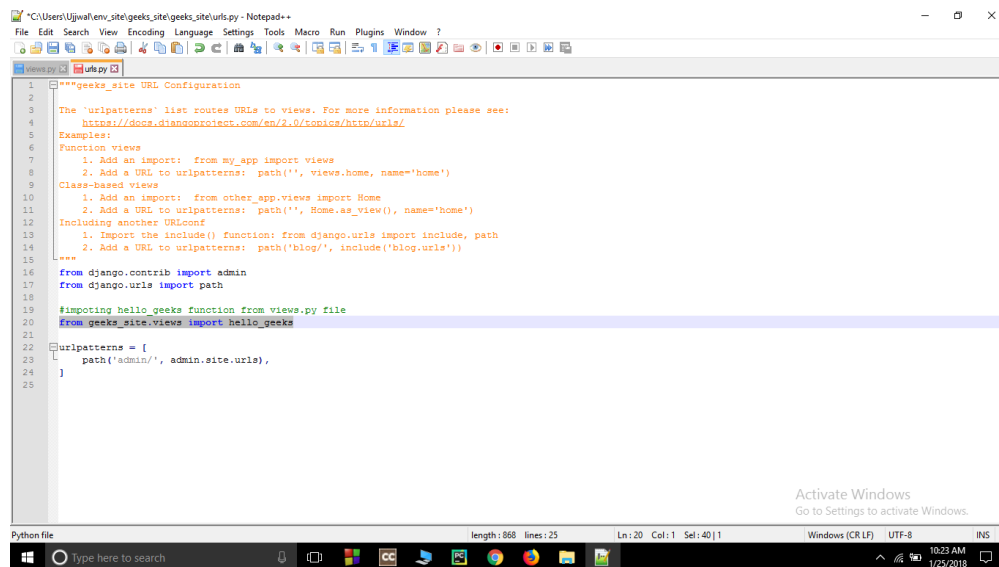
    # This will return Hello Geeks
    # string as HttpResponse
    return HttpResponse("Hello Geeks")
```

Attached Screenshot of above code –



Open `urls.py` inside project folder (`geeks_site`) and add your entry-

```
from geeks_site.views import hello_geeks
```



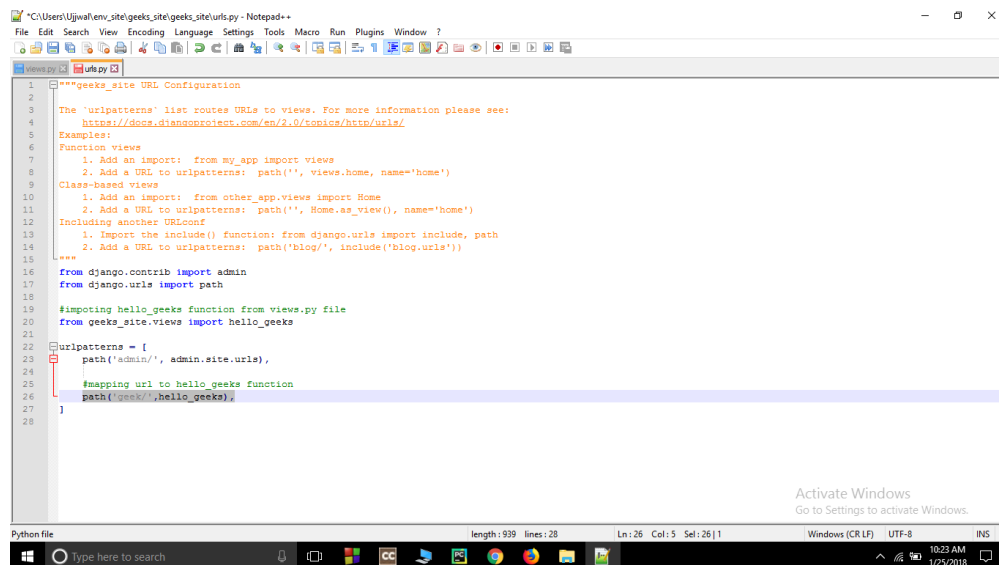
```

1  """geeks_site URL Configuration
2
3  The 'urlpatterns' list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/2.0/topics/http/urls/
5  Examples:
6  Function views
7      1. Add an import:  from my_app import views
8      2. Add a URL to urlpatterns:  path('', views.home, name='home')
9  Class-based views
10     1. Add an import:  from other_app.views import Home
11     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
12  Including another URLconf
13     1. Import the include() function: from django.urls import include, path
14     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15
16  """
17  from django.contrib import admin
18  from django.urls import path
19
20  #importing hello_geeks function from views.py file
21  from geeks_site.views import hello_geeks
22
23  urlpatterns = [
24      path('admin/', admin.site.urls),
25  ]

```

Add an entry in url field inside url patterns-

`path('geek/',hello_geek),`



```

1  """geeks_site URL Configuration
2
3  The 'urlpatterns' list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/2.0/topics/http/urls/
5  Examples:
6  Function views
7      1. Add an import:  from my_app import views
8      2. Add a URL to urlpatterns:  path('', views.home, name='home')
9  Class-based views
10     1. Add an import:  from other_app.views import Home
11     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
12  Including another URLconf
13     1. Import the include() function: from django.urls import include, path
14     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15
16  """
17  from django.contrib import admin
18  from django.urls import path
19
20  #importing hello_geeks function from views.py file
21  from geeks_site.views import hello_geeks
22
23  urlpatterns = [
24      path('admin/', admin.site.urls),
25      #mapping url to hello_geeks function
26      path('geek/', hello_geeks),
27  ]
28

```

Now to run the server follow these steps-

Open command prompt and change directory to env_site by this command-

`$ cd env_site`

Go to Script directory inside env_site and activate virtual environment-

```
$ cd Script
```

```
$ activate
```

Return to the env_site directory and goto the project directory-

```
$ cd ..
```

```
$ cd geeks_site
```

Start the server- Start the server by typing following command in cmd-

```
$ python manage.py runserver
```

Checking – Open the browser and type this url-

<http://127.0.0.1:8000/geek/>



U ujjwa...



3

[Previous Article](#)

[Next Article](#)

7 Cool Python Project Ideas for Intermediate Developers

Similar Reads

Django project - Creating a Basic E-commerce Website for Displaying...

Project Title - Basic Ecommerce Website using Django Django is a powerful framework based on python. Here we will see how to create a basic e-commerc...

3 min read

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our django...

2 min read

Django - Creating Apps | Set - 1

Prerequisites: Django – Dealing with warnings Why do we need apps? In Django Set 2 (Creating a Project), we saw how we can display text in our browser using...

1 min read

Django - Creating apps | Set - 2

In the previous article, we discussed why apps are important in Django project management? What are the benefits of using Django apps? In this article, we wi...

2 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

Creating custom user model API extending AbstractUser in Django

Every new Django project should use a custom user model. The official Django documentation says it is “highly recommended” but I’ll go a step further and say...

4 min read

Django - Creating a 404 Error Page

The 404 error is raised when the URL called upon doesn't exist or has not been defined yet. This is commonly referred to as Page does not exist error. To handle...

4 min read

Django - Creating a Multi-Page Website

Django is a python based open-source framework used for web development. Django allows the usage of the M-T-V (Model-Templates-View) architectural...

8 min read

Creating multiple user types and using proxy models in Python Django

In this article, we will understand the concept of the proxy model and the implementation of multiple user types depending on the requirements of your...

9 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

Article Tags : [GBlog](#) [Python](#) [Web Technologies](#) [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

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