Django Basics

Last Updated : 05 Sep, 2024

Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you normally will find with other frameworks. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use.

Why Django?

- Django is a rapid web development framework that can be used to develop fully fleshed web applications in a short period of time.
- It's very easy to switch database in Django framework.
- It has built-in admin interface which makes easy to work with it.
- Django is fully functional framework that requires nothing else.
- It has thousands of additional packages available.
- It is very scalable. For more visit [When to Use Django? Comparison with other Development Stacks ?](#)

Django architecture

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application. MVT Structure has

the following three parts – **Model**: Model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres). **View**: A Django View is a Python function or class that receives a web request, processes it according to defined logic (often involving interaction with Models), and returns a web response. It acts as the interface between the Model (data) and the Template (presentation), handling the flow of data and determining how to respond to different requests within a web application. **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. To check more about Django's architecture, visit [Django Project MVT Structure](#)

Installation of Django

- Install python3 if not installed in your system (according to configuration of your system and OS) from [here](#) . Try to download the latest version of python it's python3.6.4 this time.
- **Install pip-** Open command prompt and enter following command-
- **Install virtual environment-** Enter following command in cmd-
- **Set Virtual environment-** Setting up the virtual environment will allow you to edit the dependency which generally your system wouldn't allow. Follow these steps to set up a virtual environment-
  1. Create a virtual environment by giving this command in cmd-

```
python -m virtualenv env_site
```

1. Change directory to env_site by this command-

```
cd env_site
```

1. Go to Script directory inside env_site and activate virtual environment-

```
cd Scripts
```

```
activate
```

- **Install Django-** Install django by giving following command-

```
pip install django
```

- To initiate a project of Django on Your PC, open Terminal and Enter the following command

```
django-admin startproject projectName
```
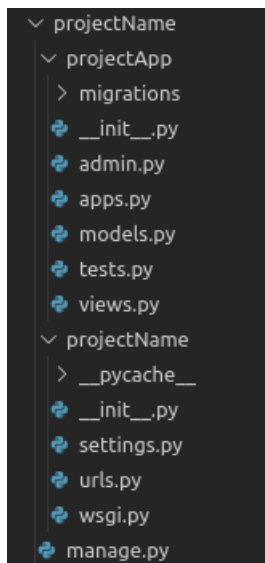
- A New Folder with name projectName will be created. To enter in the project using terminal enter command

```
cd projectName
```

- To create a basic app in your Django project you need to go to directory containing manage.py and from there enter the command :

```
python manage.py startapp projectApp
```

- Now you can see your directory structure as under :

- To consider the app in your project you need to specify your project name in INSTALLED_APPS list as follows in settings.py:

Python

```
      # Application definition
2
      INSTALLED APPS    [
```

```
5         'django.contrib.auth',
6         'django.contrib.contenttypes',
7         'django.contrib.sessions',
8         'django.contrib.messages',
9         'django.contrib.staticfiles',
10        'projectApp'
11    ]
```

- So, we have finally created an app but to render the app using urls we need to include the app in our main project so that urls redirected to that app can be rendered. Let us explore it. Move to projectName-> projectName -> urls.py and add below code in the header

```
from django.urls import include
```

- Now in the list of URL patterns, you need to specify app name for including your app urls. Here is the code for it –

Python

```python
1   from django.contrib import admin
2   from django.urls import path, include
3
4   urlpatterns = [
5       path('admin/', admin.site.urls),
6       # Enter the app name in following syntax for this to
    work
7       path('', include(&quot;projectApp.urls&quot;)),
8   ]
```

- Now You can use the default MVT model to create URLs, models, views, etc. in your app and they will be automatically included in your main project.

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our Mastering Django Framework - Beginner to Advanced Course on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N   Nave...   + Follow   161

Previous Article   **Next Article**

Django Tutorial | Learn Django Framework   Django Installation and Setup

Similar Reads

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 djang...

2 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

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

How to Fix Django "ImportError: Cannot Import Name 'six' from...

After Django 3.0, django.utils.six was removed. The "ImportError: Cannot Import Name 'six' from 'django.utils'" error occurs because of the dependency issue. Thi...

3 min read

Basics of Discrete Event Simulation using SimPy

SimPy is a powerful process-based discrete event simulation framework written in Python. Installation : To install SimPy, use the following command - pip instal...

3 min read

Basics Of Python Modules

A library refers to a collection of modules that together cater to a specific type of needs or application. Module is a file(.py file) containing variables, class...

3 min read

Selenium Python Basics

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test...

5 min read

How to Learn Python Basics With ChatGPT

Python is one of the most popular programming languages, known for its simplicity and versatility. Whether you're a complete beginner or an experienced...

4 min read

Python 3 basics

Python was developed by Guido van Rossum in the early 1990s and its latest version is 3.11.0, we can simply call it Python3. Python 3.0 was released in 200...

10 min read

Article Tags :   Python   Django-basics   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