



How to Create an App in Django ?

Last Updated : 22 Dec, 2022

Prerequisite – [How to Create a Basic Project using MVT in Django?](#)

Django is famous for its unique and fully managed app structure. For every functionality, an app can be created like a completely independent module. This article will take you through how to create a basic app and add functionalities using that app.

For example, if you are creating a Blog, Separate modules should be created for Comments, Posts, Login/Logout, etc. In Django, these modules are known as apps. There is a different app for each task.

Benefits of using Django apps –

- Django apps are reusable i.e. a Django app can be used with multiple projects.
- We have loosely coupled i.e. almost independent components
- Multiple developers can work on different components
- Debugging and code organization is easy. Django has an excellent debugger tool.
- It has in-built features like admin pages etc, which reduces the effort of building the same from scratch

Pre-installed apps –

Django provides some pre-installed apps for users. To see pre-installed apps, navigate to projectName -> projectName -> settings.py

In your settings.py file, you will find INSTALLED_APPS. Apps listed in INSTALLED_APPS are provided by Django for the developer's comfort.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Also, Visit : [Django ORM – Inserting, Updating & Deleting Data](#)

Creating an App in Django :

Let us start building an app.

Method-1

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

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

Method-2

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

```
django-admin startapp projectApp
```

- Now you can see your directory structure as under :

```

  ▼ projectName
    ▼ projectApp
      > migrations
      + __init__.py
      + admin.py
      + apps.py
      + models.py
      + tests.py
      + views.py
    ▼ projectName
      > __pycache__

```

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

```

+ urls.py
+ wsgi.py
+ manage.py

```

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

Python3

Application definition

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'projectApp'
]

```

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 the app name for including your app URLs. Here is the code for it –

Python3

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    # Enter the app name in following
    # syntax for this to work
    path('', include("projectApp.urls")),
]
```

- 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.

The main feature of Django Apps is independence, every app functions as an independent unit in supporting the main project.

Now the urls.py in the project file will not access the app's url.

To run your Django Web application properly the following actions must be taken:-

1. Create a file in the apps directory called urls.py
2. Include the following code:

Python3

```
from django.urls import path
#now import the views.py file into this code
from . import views
urlpatterns=[
    path('',views.index)
]
```

The above code will call or invoke the function which is defined in the views.py file so that it can be seen properly in the Web browser. Here it is assumed that views.py contains the following code :-

Python3

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello Geeks")
```

After adding the above code, go to the settings.py file which is in the project directory, and change the value of ROOT_URLCONF from 'project.urls' to 'app.urls'

From this:-

```
ROOT_URLCONF = 'project.urls'
```

To this:

```
ROOT_URLCONF = 'app.urls'
```

3. And then you can run the server(127.0.0.1:8000) and you will get the desired output

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!

Previous Article

Python Web Development With Django

Next Article

Weather app using Django | Python

Similar Reads

Create Word Counter app using Django

In this article, we are going to make a simple tool that counts a number of words in text using Django. Before diving into this topic you need to have some basic...

3 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

Create Newsletter app using Mailchimp and Django

In this article, we'll use MailChimp and Django to make a simple mailing app. When people subscribe to our app, we'll learn how to create an audience list an...

5 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 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

Python | Django News App

Django is a high-level framework which is written in Python which allows us to create server-side web applications. In this article, we will see how to create a...

2 min read

Weather app using Django | Python

In this tutorial, we will learn how to create a Weather app that uses Django as backend. Django provides a Python Web framework based web framework that...

2 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

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