**Complete Step-up of Your First Django Project In 13 Simple Steps**

*Every Django project have some essential steps to follow to complete it. Here are the 13 steps to create and setup your first Django Project and Application, which include creating project, app, database, templates and configuring other basic commodities. So, next time you want to create a project you come here and perform these steps, it’s like your notes, no need to remember all the processes.*



Before you start building a new Django web application, you always need to complete a couple of steps. These steps are the backbone of your every Django project, you can’t complete any of your project without performing them. This tutorial gives you a brief guide for the important steps to set up a new Django project.

One more thing, please note. If you have understanding of building static web pages with HTML, CSS, JAVASCRIPT or with BOOTSTRAP, then after these steps you will be able to build your basic Django website. Before starting you have to make sure these important necessities.

* [Install Python](https://realpython.com/installing-python/)
* [Install Django](https://docs.djangoproject.com/en/3.2/topics/install/)

Before we start building our project, you have to understand the two most important terms Django Project and Django App and difference between them.

* **A Django project** is a high-level unit of organization that contains logic which governs whole web application. *And a project can contain multiple apps.*
* **A Django app** is a lower-level unit of web application. You can have one to many apps in a project, but one app is must require.

**Example:** Your whole website is your project and blog, shop and other functionalities in it are the apps of that project.

Now lets move on our steps to setup Django Project…

**1. Create New Django Project**

Go to your code editor (like Visual Studio Code, Pycharm, etc…) open your project folder, where you want to create project. Now open terminal of the code editor and run the following command. (For this tutorial I’ll use “*helloworld*” as an example for the project name.)

django-admin startproject helloworld

This command will create a default folder structure, which includes some Python files and your management app that has the same name as your project (*helloworld*)

**2. Enter Into Your Project**

Now type the following command in the terminal. It will enter you in the project folder, where we will perform further functionalities.

cd helloworld

*Your project is created, you can open your project by running (*python manage.py runserver)*command in the terminal and visit IP address given in terminal, it’s probably something like this (*[*http://127.0.0.1:8000/*](http://127.0.0.1:8000/)*).*

**3. Create Your First Django App**

We already created our project so, it’s time for creating our Django app which should be inside of the project. For app creation we need to run following command in our terminal, while we are inside of the project folder. (I am using ‘blog’ name as my app, you can use whatever you want.)

python manage.py startapp blog

**4. Register Your App**

Now you have to register you created app in project. For doing that you have to open your settings.py file (which is inside of your project folder). Now scroll down it to INSTALLED\_APPS section, and add your app name blog inside it (with inverted commas) on the top or bottom and leave the other code as it is.

INSTALLED\_APPS = [  
'blog',  
.  
.  
.  
]

**5. Create Static & Templates Folders**

Now lets create two very important folder where your most of the files will store.

**Static Folder** will contain your static files (like images, videos, text files, CSS or JavaScript files, etc…). Remember these will be accessible to others/visitors, not very secure.

**Templates Folder** will contain your dynamic files (like html files, site code and other important files). These files are the more secure files and not directly accessible to others.

Now open your project folder in and create static and templates folders. You have no need of commands, just create like you create new folders.

**6. Set Static Directories**

After creating those folder now come back to your code editor and set their directories. First you have to set static directory. Go to your settings.py file. At very beginning were all other libraries are imported let import os module in it by typing...

import os

Now scroll settings.py down to the end and type following code in it.

STATICFILES\_DIRS = [  
os.path.join(BASE\_DIR, 'static')  
]

**7. Set Template Directories**

Well it’s for setting up the next directory which is Templates. In same settings.py file find TEMPLATES in this section look for 'DIRS':[], and in those empty brackets type the following code in it and leave the other code as it is. You just have to edit 'DIRS' field.

TEMPLATES = [{  
.  
'DIRS': [os.path.join(BASE\_DIR, 'templates')],  
.  
.  
.  
}]

*If this is too much for you, Go and do Relax yourself for a minute.*



**8. Create Templates**

Templates are the web pages like index.html , about.html or contact.html . Templates can be a portion of a web page like navbar or header or footer . We can extends those portions in our web pages like {%extends 'navbar.html'%} . And for including styles or scripts in out templates from static folder we will use following code.

<link rel="stylesheet" href="/static/css/styles.css">  
<script src="/static/js/scripts.js"></script>

*You can learn more about it online. Here is a useful source for betting understanding.*[Visit Here](https://www.geeksforgeeks.org/django-templates/)

**9. Set URLs of Project & App**

Now let’s set configure URLs of our project so which template will show on which URL. First we will configure URLs in Project, for this open your urls.py file from project folder. Import include module from django.urls

from django.urls import path, include

Now lets configure the path of our app, for this add a path in urlpatterns after admin path

urlpatterns = [  
path('admin/', admin.site.urls),  
path('', include('home.urls'),  
]

After project URLs we will configure app URLs. First create urls.py files in your app blog folder and then type the following code in it.

from django.urls import path, include  
from django.contrib import admin  
from blog import views  
urlpatterns = [  
path('', views.index, name='blog'),  
]

**10. Define & Render URLs**

Let’s now render our URLs in views.py file. Open it and write this code in it.

from django.shortcuts import render  
def index(request):  
 return render(request, 'index.html', {})

**11. Create A Super-User**

Django offers you a built-in very powerful admin panel, to access it you have to create a superuser which ill be able to access every feature of that admin panel. (**Remember**: there few other types of users on the basis of there roles, you can explore them by [click here](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Authentication).) For creating superuser in Django you have to open your terminal and type the following command in it. (I am using admin as my superuser)

python manage.py createsuperuser admin  
Email:  
Password:  
(Re-type) Password:

For Email you will type an email address and in password type a strong password (**Remember**: when you type password it won’t show it to you so, just type it and hit Enter key) and Re-type it. After this you might see a message so, just enter y and hit Enter key.

*You superuser is just created, you can now open your admin panel by visiting*[*http://127.0.0.1:8000/admin/*](http://127.0.0.1:8000/admin/)*link. (But first you must run your server with runserver command)*

**12. Run Migrations**

You are almost finish one more step which very import and you might realize it that if you try to access the Project URL and it through an error or migrations in your terminal. So, now we will fix it by run migrations in our project.

***Note****: there are some important tables and configuration/changes which we perform in our settings.py and other files. So, makemigrations will create a file of your changes and store them.*

Go to your terminal and type…

python manage.py makemigrations

**Note**: like changes there are important tables needed to execute in our database. For this purpose we will migrate those tables with migrate command.

So, go to your terminal and type…

python manage.py migrate

**13. Run Project**

Finally we just finish all the important steps of our Django project setup. So, at the end it’s time to run our project and see how it looks. Now type the following command in your terminal…

python manage.py runserver

If all clears, visit [http://127.0.0.1:8000](http://127.0.0.1:8000/admin/)/ link and open your first Django web application, you can visit your admin panel by visiting <http://127.0.0.1:8000/admin/>

**Final Thoughts**

While reading all this you notice that we work a lot on manage.py file, it is the most important file of our Django project so, make sure don’t effect it. I know there are lot more to learn about Django and you might not understand some of the terms, so, please explore them online and understand them. Here is the [Django Documentation](https://docs.djangoproject.com/en/3.2/), the most useful resource.

Thank you people for reading this content I’m hopping that you guys find it useful. If there is anything you want to ask or add in it, please let me know in comment section and I’ll reach you ASAP.