



How to Send Email with Django

Last Updated : 04 Sep, 2024

Django, a high-level [Python](#) web framework, provides built-in functionality to send emails effortlessly. Whether you're notifying users about account activations, sending password reset links, or dispatching newsletters, Django's robust email handling system offers a straightforward way to manage email communication. This tutorial will guide you through the process of setting up and sending emails in Django, covering configuration, templates, and practical examples, ensuring that you can integrate email functionality into your [Django applications](#) with ease.

Django to send emails with SMTP

Consider a project named geeksforgeeks having an app named geeks. Refer [this](#) to create Django projects and apps. Now let's demonstrate this in geeksforgeeks project. In your "geeks" app's **settings.py** file, add the following code:

Python



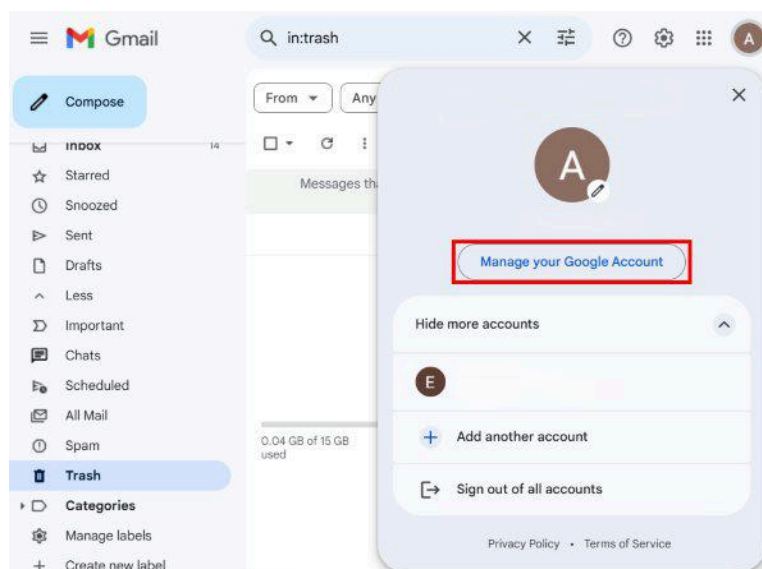
```
1 EMAIL_BACKEND =  
    'django.core.mail.backends.smtp.EmailBackend'  
2 EMAIL_HOST = 'smtp.gmail.com'  
3 EMAIL_USE_TLS = True  
4 EMAIL_PORT = 587  
5 EMAIL_HOST_USER = #sender's email-id  
6 EMAIL_HOST_PASSWORD = #password associated with above email-  
    id (not the regular password)
```

```
settings.py  x  signup.html  x
115
116 USE_TZ = True
117
118
119 # Static files (CSS, JavaScript, Images)
120 # https://docs.djangoproject.com/en/3.0/howto/static-files/
121
122 STATIC_URL = '/static/'
123 EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
124 EMAIL_HOST = 'smtp.gmail.com'
125 EMAIL_USE_TLS = True
126 EMAIL_PORT = 587
127 EMAIL_HOST_USER = 'xabc6457@gmail.com'
128 EMAIL_HOST_PASSWORD = 'xyz123abc@'
129
```

By default, Google does not allow third party apps to use mail services using our email address and actual password. So, we have to create an app password which will be only utilized by our Django mail application. In order to generate an app password for your google account, follow these steps:

Step 1

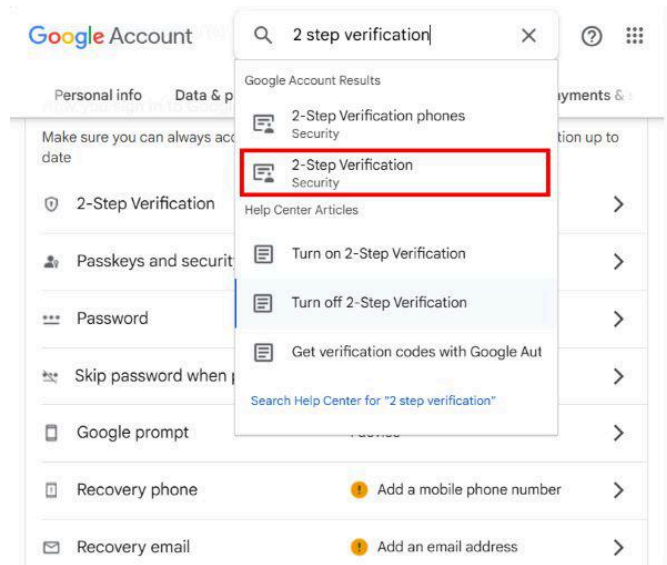
Sign in to your google account and click on Manage account.



Step 1

Step 2

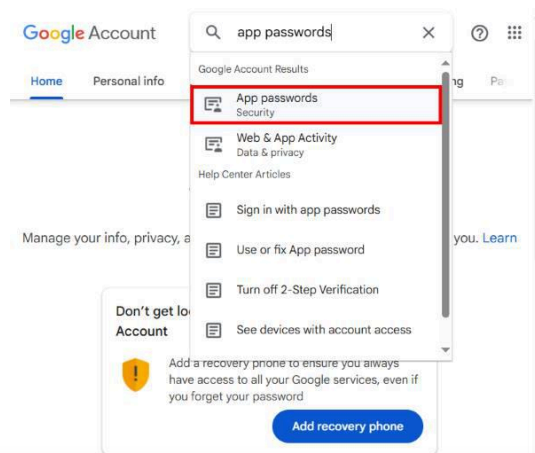
Enable two factor authentication by typing “2 step verification” on the search bar.



Step 2

Step 3

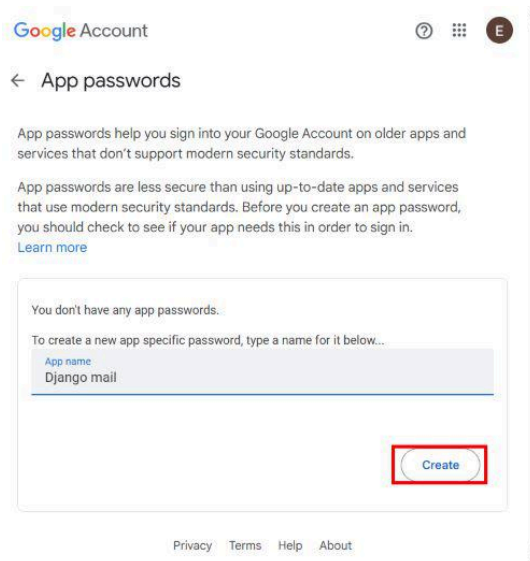
After enabling 2FA, type “App passwords” on the search bar.



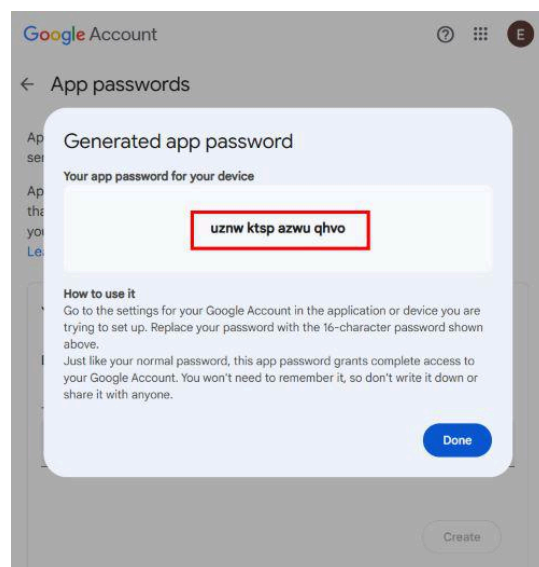
Step 3

Step 4

Enter a project name and your unique password will be provided to use it in your project. Note that the password is only shown once, and it is valid until it is manually removed from the app passwords tab. Hence copy the password when it is shown and use it as EMAIL_HOST_PASSWORD in the Django settings.py file.



Step 4 (i)



Step 4 (ii)

The views.py file for this application contains the following code:

Python

```

1  from django.shortcuts import render
2  from django.http import HttpResponse
3  from django.core.mail import send_mail
4  from django.conf import settings
5
6  def send_mail_page(request):
7      context = {}
8
9      if request.method == 'POST':
10         address = request.POST.get('address')
```

```
12         subject = request.POST.get('subject')
13         message = request.POST.get('message')
14
15         if address and subject and message:
16             try:
17                 send_mail(subject, message,
18                           settings.EMAIL_HOST_USER, [address])
19                 context['result'] = 'Email sent
20                 successfully'
21             except Exception as e:
22                 context['result'] = f'Error sending email:
23                 {e}'
24
25         else:
26             context['result'] = 'All fields are required'
27
28     return render(request, "index.html", context)
```

Now we will understand what exactly is happening. Here the `send_mail()` is an inbuilt Django function which takes 4 important arguments,

- `subject` refers to the email subject.
- `message` refers to the email message, the body of the email.
- `email_from` refers to the sender's details. This takes the `EMAIL_HOST_USER` from `settings.py` file, where you added those lines of code earlier.
- `recipient_list` is the list of recipients to whom the mail has to be sent.

The return value of this function is either 0 or 1 depending upon the number of messages sent.

Other optional arguments of the `send_mail()` function includes,

- `fail_silently`: It is a boolean value used to raise an `smtpplib.SMTPException` if an error occurs.
- `auth_user`: The optional username to use to authenticate to the SMTP server. If this isn't provided, Django will use the value of the `EMAIL_HOST_USER` setting.
- `auth_password`: The optional password to use to authenticate to the SMTP server. If this isn't provided, Django will use the value of the `EMAIL_HOST_PASSWORD` setting.

- connection: The optional email backend used to send the mail. If unspecified, an instance of the default backend will be used.
- html_message: It is used to send HTML pages in the mail body rather than sending plain text.

The index.html looks like:

HTML

```
1 {% load static %}
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8     <link rel="stylesheet" href="{% static 'style.css' %}" />
9     <title>Mail App</title>
10 </head>
11 <body>
12     <div>
13         <h1>Django Mail App</h1>
14         <form method="post">
15             {% csrf_token %}
16             <label for="address">To:</label>
17             <input id="address" type="email"
18             name="address"/>
19             <label for="subject">Subject:</label>
20             <input id="subject" name="subject"/>
21             <label for="message">Message:</label><br>
22             <textarea id="message" name="message">
23             </textarea>
24             <input type="submit">
25         </form>
26         <p>{{ result }}</p>
27     </div>
28 </body>
```

```
</html>
```

And the styles.css looks like:

CSS

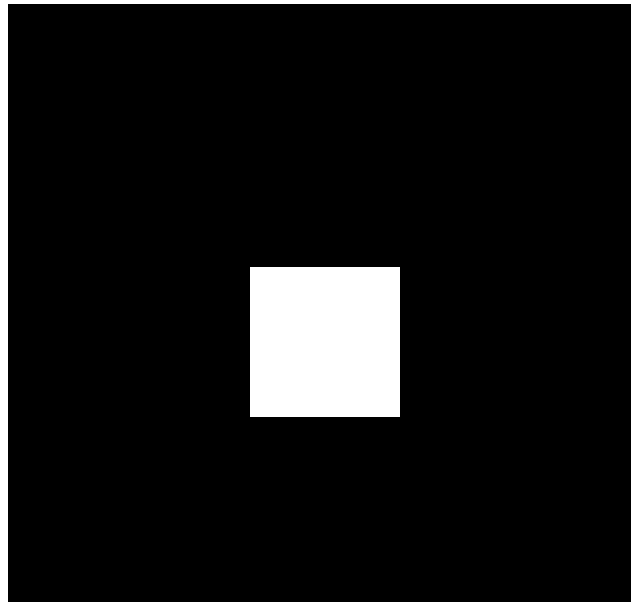
```
1  h1{
2      margin: 10px auto 20px auto;
3      color: green;
4      display: block;
5  }
6  input{
7      width: 300px;
8      height: 25px;
9      display: block;
10     margin-bottom: 20px;
11 }
12 div{
13     display: flex;
14     flex-direction: column;
15     align-items: center;
16 }
17 textarea{
18     resize:none;
19     width: 302px;
20     height: 100px;
21     margin-bottom: 10px;
22 }
23 input[type="submit"]{
24     width:150px;
25     height: 30px;
26     background-color: green;
27     color:white;
28     border: none;
29     margin: 0 auto;
30 }
```

Add the following in urls.py:

Python

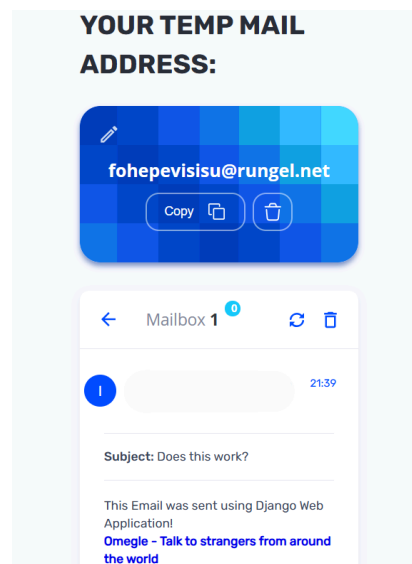
```
1 from django.contrib import admin
2 from django.urls import path
3 from mailsender import views
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('', views.send_mail_page)
8 ]
```

Output



00:00

00:21

*Mail received*

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!

s srival...



23

Previous Article

ToDo webapp using Django

Next Article

Django project to create a Comments System

Similar Reads

Send PDF File through Email using pdf-mail module

pdf_mail module is that library of Python which helps you to send pdf documents through your Gmail account. Installing Library This module does not come built-...

2 min read

Send Email Using yagmail in Python

In this article, we are going to see how to send email using yagmail. yagmail(Yet Another Gmail) is a module in python which used to send emails using Python....

3 min read

Email + Social Logins in Django - Step by Step Guide

This article revolves around a Django Project. It includes Email + Social Login integration in any Django project. We have used React as front end to...

14 min read

Django Sign Up and login with confirmation Email | Python

Django by default provides an authentication system configuration. User objects are the core of the authentication system. Today we will implement Django's...

7 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

Send SMS updates to mobile phone using python

If you are running any python script and want to send regular updates from your script to your mobile phone through SMS, you can use SinchSMS API to send...

2 min read

Article Tags : [Project](#) [Python](#) [Django-Projects](#) [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](#)

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](#)

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

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

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

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