How to automate sending emails using python 2024-2025

Issam · Follow

3 min read · Nov 1, 2024

1

Maybe you’ve found hundreds of articles or tutorials about this topic, but in almost all the cases:

- The tutorial is outdated
- it doesn’t work as intended
- it doesn’t support all types of emails
- or something else…

In this article you’ll find a method that is guaranteed to be functional if you use it during the mentioned years.

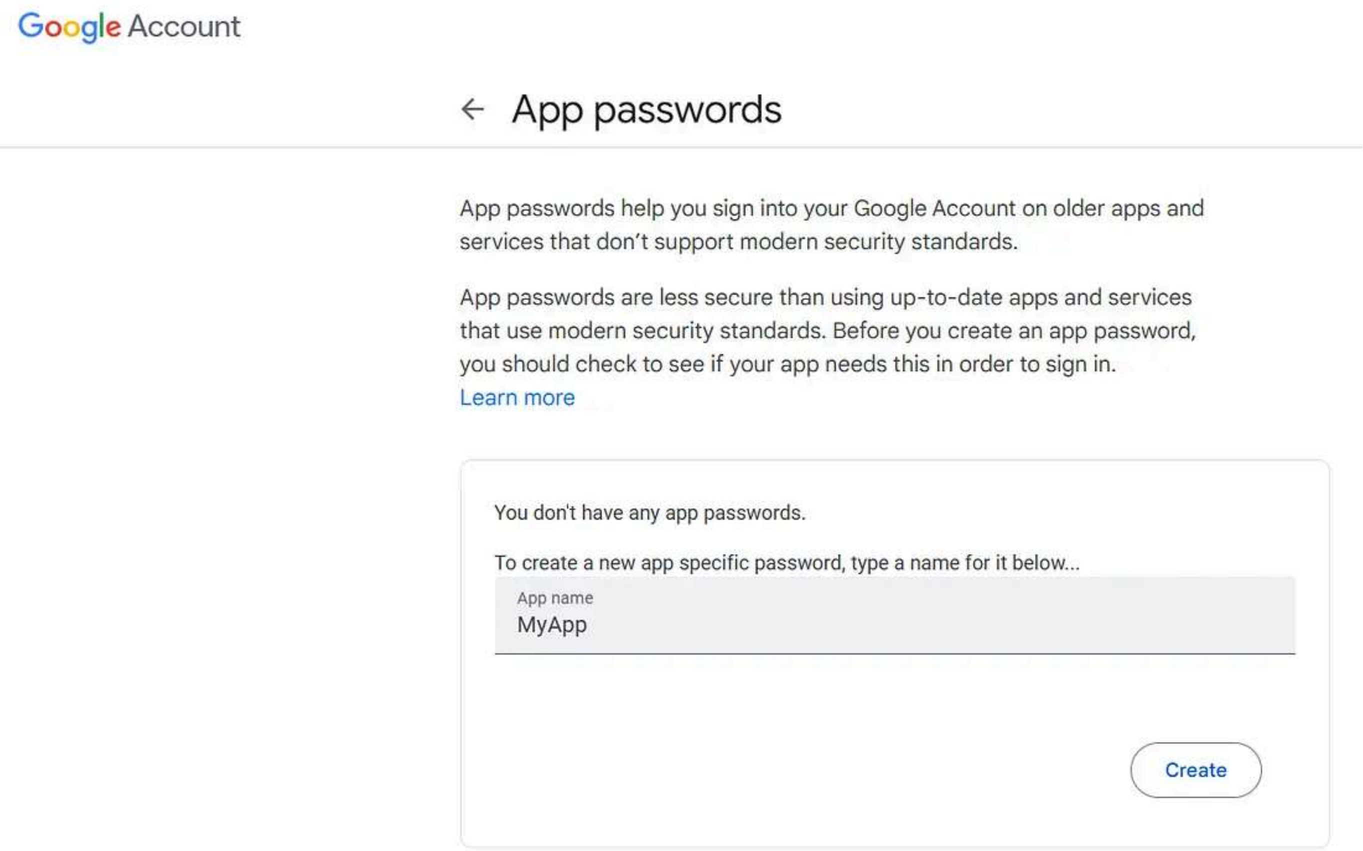
For this to work you’ll need the following things:

- A Gmail account

- A python environment (with basic knowledge of programming)
- 20 minutes of your time

1 — Get a 16-character password from Gmail

First of all, activate 2 step authentication, then visit this link to create an app password, it would look something like this:


Google Account

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.
Learn more

You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name: MyApp

2 — Create this project structure:

create a new folder, in the root create the following files:

.env | main.py | usage.py | index.html

3 — Implement main.py:

install the required libraries using the command:

```
pip install secure-smtplib email
```

```
import smtplib
import ssl
from email.message import EmailMessage
import mimetypes
import os

# Define email sender and receiver
email_sender = os.environ.get('EMAIL_SENDER')
email_password = os.environ.get('EMAIL_PASSWORD')

# Add SSL ( of security)
context = ssl.create_default_context()

def send_email(email_receiver, subject, body, pdf_path=None, is_html=False):
    em = EmailMessage()

    em['From'] = email_sender
    em['To'] = email_receiver
    em['Subject'] = subject

    if is_html:
        em.add_alternative(body, subtype='html')
    else:
        em.set_content(body)

    # Attach PDF if provided
    if pdf_path:
        # Guess the MIME type and subtype
        mime_type, _ = mimetypes.guess_type(pdf_path)
        mime_type, mime_subtype = mime_type.split('/')

        with open(pdf_path, 'rb') as pdf_file:
            em.add_attachment(pdf_file.read(),
                              maintype=mime_type,
                              subtype=mime_subtype,
                              filename=os.path.basename(pdf_path))
    # Log in and send the email
    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
        smtp.login(email_sender, email_password)
        smtp.sendmail(email_sender, email_receiver, em.as_string())
```

Explanation:

- **The modules:** Imports `smtplib` and `ssl` for secure email sending, `EmailMessage` for email construction, `mimetypes` for file MIME types, and `os` to access environment variables.
- **Defines email sender and password:** Gets the email sender's address and password from environment variables (`EMAIL_SENDER` and `EMAIL_PASSWORD`).
- **Sets up SSL context:** Uses `ssl.create_default_context()` to create a secure SSL context for sending the email.
- **Sets up email content:** Creates an `EmailMessage` object, sets the sender, recipient, and subject. If `is_html` is `True`, sets the email body as HTML; otherwise, it's plain text.
- **Handles PDF attachment (optional):** If a `pdf_path` is provided, it guesses the MIME type of the PDF, then attaches it to the email.
- **Sends email over SSL connection:** Establishes a secure connection with Gmail's SMTP server, logs in, and sends the email.

4 — Implement the rest of the project:

.env file :

```
EMAIL_SENDER = 'Your email'
EMAIL_PASSWORD = "Your 16 characters password"
```

.html file (the email template):

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
    }
    p {
      margin: 0 0 10px;
    }
    .signature {
      font-weight: bold;
    }
    .ps {
      font-style: italic;
    }
  </style>
</head>
<body>
  <p>Hello,</p>
  <p>This is a  <strong>test</strong> email</p>
  <p>Best regards,</p>
  <p class="signature">Name</p>
  <p class="ps">P.S.: You will find the pdf attached to this email.</p>
</body>
</html>
```

usage.py (For combining all the pieces) :

```
from app import send_email

# Set the subject of the email
subject = 'Test email'

# Read the HTML content from the file
with open('index.html', 'r') as file:
    body = file.read()

email_receiver = 'skyzlim1@gmail.com'
pdf_file_path = "test.pdf"



send_email(email_receiver, subject, body, pdf_file_path, is_html=True)
```

here we just read the content of html template and then use the send_email

Open in app ↗ Sign up Sign in

Medium Search Write

you’ll find the complete code for this in this GitHub repo: https://github.com/issamelimrany/Send-email.git

If you’ve found this article useful, let me know in the comments below!

Python Automation Email Marketing Email Programming

Written by Issam

0 Followers · 0 Following

Follow

An aspiring AI engineer with hands-on experience in developing and deploying full-stack machine learning and deep learning solutions, from concept to production

No responses yet

What are your thoughts?

Respond

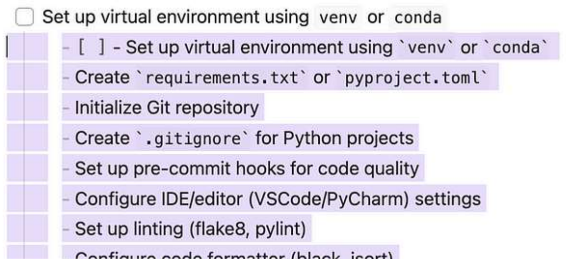
Recommended from Medium

Kevin Meneses González

How to 300x Your Productivity with These 13 AI Tools

📌 Introduction — The story of how I stopped wasting time

Feb 2 330 7



```
☐ Set up virtual environment using venv or conda
   - [ ] - Set up virtual environment using `venv` or `conda`
   - Create `requirements.txt` or `pyproject.toml`
   - Initialize Git repository
   - Create `.gitignore` for Python projects
   - Set up pre-commit hooks for code quality
   - Configure IDE/editor (VSCode/PyCharm) settings
   - Set up linting (flake8, pylint)
   - Configure code formatter (black, isort)
```

In CodeX by AI Rabbit

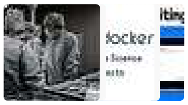
Goodbye Obsidian

5d ago 279 26

Lists

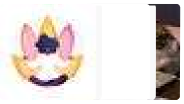
General Coding Knowledge

20 stories · 1907 saves



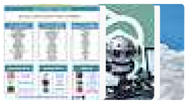
Coding & Development

11 stories · 1002 saves



Stories to Help You Grow as a Software Developer

19 stories · 1593 saves



ChatGPT

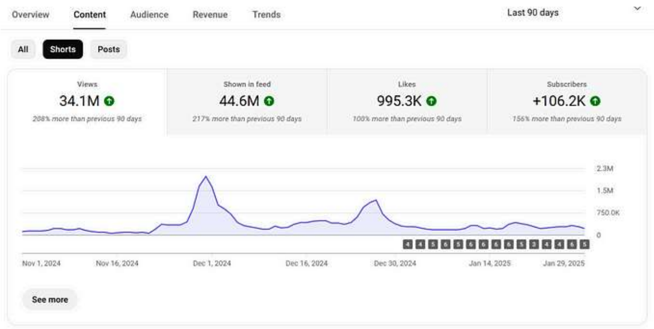
21 stories · 959 saves

In Code Like A Girl by Sivan Hermon

Micro-Retirement: A New Way To Live

Micro-Retirement: Tasting Freedom Before It’s Too Late

4d ago 👏 1.3K 💬 28



Last 90 days

Views 34.1M — 208% more than previous 90 days

Shown in feed 44.6M — 217% more than previous 90 days

Likes 995.3K — 100% more than previous 90 days

Subscribers +106.2K — 156% more than previous 90 days

In Write A Catalyst by Awais Shahid

How Much YouTube Paid Me For 320 Shorts in 3 Months

Expectations vs reality

✦ Feb 1 👏 4.1K 💬 129

In AI Simplified in Plain English by Manpreet Singh

Making a RAG App with DeepSeek-R1 & Ollama

Source: canva

In Dev Genius by Nirav Parmar

How to Use Google Sheets as a Web API 📊 ✨

Learn how to use Google Sheets as a web API to manage and interact with your data…

✦ 5d ago 👏 102 💬 4

✦ Aug 16, 2024 👏 1

See more recommendations