# Automated Daily Email of Top 10 Hacker News Headlines using GitHub Actions

## Overview

This guide explains how to scrape the top 10 headlines from The Hacker News website and send them via email daily at 9am using GitHub Actions. It includes the Python script, GitHub Actions workflow configuration, secrets setup, and troubleshooting tips.

## Step-by-Step Setup

### 1. Create GitHub Repository

Go to https://github.com and create a new repository (e.g., daily-hackernews-email). Clone it locally or use the web editor.

### 2. Add Python Script

Create a file named send\_hackernews.py and paste the following code:

import requests

from bs4 import BeautifulSoup

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

# Gmail credentials

GMAIL\_USER = ‘yourEmail’

GMAIL\_APP\_PASSWORD = 'Password'

TO\_EMAIL = 'yourEmail'

# Scrape headlines

def get\_headlines():

url = 'https://thehackernews.com'

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

articles = soup.find\_all('a', class\_='story-link')[:10]

headlines = []

for article in articles:

title\_tag = article.find('h2')

if title\_tag and article.has\_attr('href'):

title = title\_tag.text.strip()

link = article['href']

headlines.append(f'<li><a href="{link}">{title}</a></li>')

if headlines:

return '<ul>' + ''.join(headlines) + '</ul>'

else:

return '<p>No headlines found.</p>'

# Send email

def send\_email(content):

msg = MIMEMultipart('alternative')

msg['Subject'] = 'Top 10 Hacker News Headlines'

msg['From'] = GMAIL\_USER

msg['To'] = TO\_EMAIL

html = f"""\

<html>

<body>

<h3>Top 10 Headlines from The Hacker News</h3>

{content}

</body>

</html>

"""

msg.attach(MIMEText(html, 'html'))

server = smtplib.SMTP('smtp.gmail.com', 587)

server.starttls()

server.login(GMAIL\_USER, GMAIL\_APP\_PASSWORD)

server.sendmail(GMAIL\_USER, TO\_EMAIL, msg.as\_string())

server.quit()

# Run

if \_\_name\_\_ == '\_\_main\_\_':

headlines\_html = get\_headlines()

send\_email(headlines\_html)

### 3. Configure GitHub Secrets

Go to your repository → Settings → Secrets and variables → Actions → New repository secret. Add the following secrets:  
- GMAIL\_USER: your Gmail address  
- GMAIL\_APP\_PASSWORD: your Gmail app password (generated from <https://myaccount.google.com/apppasswords>)

### 4. Create GitHub Actions Workflow

Create a file at .github/workflows/daily\_email.yml with the following content:

name: Daily Hacker News Email  
  
on:  
 schedule:  
 - cron: '0 1 \* \* \*' # 9am Malaysia time (UTC+8)  
 workflow\_dispatch:  
  
jobs:  
 send\_email:  
 runs-on: ubuntu-latest  
 steps:  
 - name: Checkout repo  
 uses: actions/checkout@v3  
  
 - name: Set up Python  
 uses: actions/setup-python@v4  
 with:  
 python-version: '3.x'  
  
 - name: Install dependencies  
 run: pip install requests beautifulsoup4  
  
 - name: Run script  
 env:  
 GMAIL\_USER: ${{ secrets.GMAIL\_USER }}  
 GMAIL\_APP\_PASSWORD: ${{ secrets.GMAIL\_APP\_PASSWORD }}  
 run: python send\_hackernews.py

### 5. Commit and Push

Run the following commands to push your changes to GitHub:

git add .  
git commit -m "Initial commit"  
git push origin main

### 6. Manual Test

To test manually:  
- Go to your repo → Actions tab  
- Select the workflow → Click 'Run workflow'  
- Check logs and verify email delivery

## Troubleshooting

- SMTPAuthenticationError: Check Gmail App Password and ensure 2FA is enabled  
- No headlines in email: Update scraping logic to use 'a.story-link' and 'h2' tags  
- Secrets exposed: Ensure credentials are stored in GitHub Secrets and not hardcoded

## Security Note

You can keep the repository public as long as secrets are stored securely in GitHub Secrets. Do not hardcode your Gmail credentials in the script. If unsure, set the repository to private.