

Newsletter of Newsletters

Objective:

The main goal of this project is to help CEO's and all the higher hierarchical employees to stay relevant in what is happening in their industry.

Explanation:

All of us have subscribed to newsletters which we find relevant for us. So does all the higher hierarchical employees(CEO's, CTO's, CFO's). But, will they go through all the newsletters?. It's a big question. **The newsletter contains a summary of all the important news happening in that blog.** So I came up with the solution, Newsletter of Newsletters. If we add the links of webpages to the python code I have created it will collect the newsletters of all the webpages and ranks them based on our priority and sends an email to us generating a new newsletter containing only the important news from all the newsletters. The beauty of the application is that it doesn't store the data as of now anywhere. So no storage is needed. All the process is automated from **scrapping news, summarizing text, creating email and sending an email.**

I'll start by explaining the working of the application.

RPA Packages Used:

```
1)requests          --- #Requesting the URL
2)bs4                --- #Web Scrapping
3)validator_collection --- #Validator the conditions
4)git+"https://github.com/code-lucas/newspaper.git" --- Extracting
newsletter contents
5)goose3             --- #Webwrapper Package
6)jinja2             --- #Creating a newsletter template
7)smtplib            --- #Sending email
```

Azure Service:

Azure Web Jobs can be used to schedule this python program to run weekly once.

Final Result Newsetter:(Screenshot attached)



How to run this application?

Upload the **Newsletter_of_Newsletters.ipynb** in google colab or any other jupyter notebook and run all the cells it will send you a mail of the newsletter.

```
#!/usr/bin/env python

from smtplib import SMTP          # sending email
from email.mime.text import MIMEText # constructing messages

from jinja2 import Environment    # Jinja2 templating

msg = email.message.Message()
msg['Subject'] = 'Newsletter of Newsletters'

recipient_list = ["newsofnewsletter@gmail.com"]
#email_content="hi"
msg['From'] = "newsofnewsletter@gmail.com"
msg['To'] = recipient_list
password="Student#1899!"
msg.add_header('Content-Type', 'text/html')
msg.set_payload((email_content).encode('ascii', 'ignore'))

s = smtplib.SMTP('smtp.gmail.com: 587')
s.starttls()

# Login Credentials for sending the mail
s.login(msg['From'], password)

s.sendmail(msg['From'], msg['To'], msg.as_string())
```

In last cell of notebook you can find this code, where you just add the people's email in recipient list and run the program you will the email.

```
import re
import requests
from validator_collection import checkers
import dateutil.parser
from datetime import date

# Pages from which the newsletters are scrapped
## Update the pages on a weekly basis for weekly updates
Pages=["https://awsinsider.net/Home.aspx", "https://dataengweekly.substack.com/",
       "https://news.dataelixir.com/t/t-EBBA06B24152E04B2540EF23F30FEDED",
       "https://www.datascienceweekly.org/newsletters/data-science-weekly-newsletter-issue-315"]

#Sources of the newsletters
Sources=["awsinsider", "dataengweekly",
        "dataelixir", "datascienceweekly"]
```

The above pic is the part of the program where you have to add the newsletter source links.

Code explanation:

Cell 1:

First cell in the python code is the necessary packages need to run the program if you run the cell it will install the packages by itself

Cell 2:

Scrapping the pages of the newsletters

Cell 3:

Taking the necessary content from the newsletters

Cell 4 to until N-2 Cells : (N- Total No of Cells)

Summarizing the contents read from newsletters and ranking the news from newsletter based on importance

N-2th Cell:

Automatically Creating the newsletter.

Final Cell:

Sending the email.