Hire with us!   Login

- Python Tutorial
- Django Tutorial
- Python Bitwise Operators
- Defaultdict in Python
- Which Python Modules are useful for competitive programming?
- Max Heap in Python
- Pandas Tutorial
- Django Basics
- ML | Credit Card Fraud Detection
- Role of SemiColon in various Programming Languages
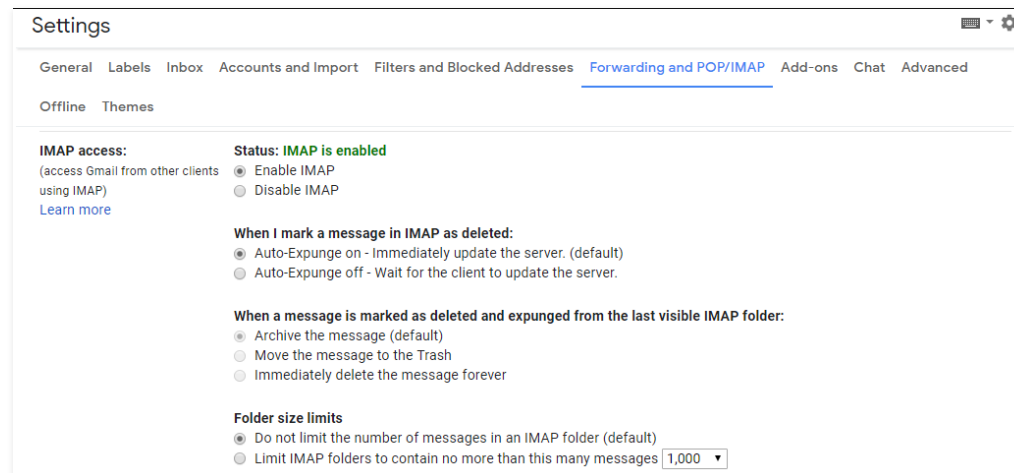- Corona Virus Live Updates for India - Using Python
- Python Itertools

Python | Fetch your gmail emails from a particular user

If you are ever curious to know how we can fetch Gmail e-mails using Python then this article is for you.

As we know Python is a multi-utility language which can be used to do a wide range of tasks. Fetching Gmail emails though is a tedious task but with Python, many things can be done if you are well versed with its usage. Gmail provides IMAP access to clients who want to access Gmail without manually logging in the browser.

In setting page, enable this before running script.

Most popular in Python

LRU Cache in Python using OrderedDict

PDFmyURL

Implementation:

The libraries used in this implementation includes **imaplib**, **email**. You have to manually go and make IMAP access enabled by going into your Gmail account settings. After this only you could access your Gmail account without logging in browser.

- Three functions are defined in the implementation which is used to get email body, search for emails from a particular user and get all emails under a label.
- For showing results I have sent email to my id from my another Gmail account. Now I will be fetching emails from my Gmail account which is received from my another Gmail account.
- The process begins from making Gmail connection with the help of *imaplib library* and proving our Gmail login credentials to it.
- After logging we are selecting emails under the label: Inbox which is a default labeled section for all users. However, you can create your own labels also.
- Then we are calling get emails function and provide it the parameter from search function result i.e "from user"
- In get emails function we are putting all emails in an array named "msgs"
- Now print to see the msgs array
- Now we can easily iterate over this array. We are iterating it in the order the emails arrived. Then we are searching for the index from where our content begins. This indexing part will be different for different emails/users and the user can manually change the indexes to print only that part which they require.
- We have our results printed out.

Below is the Python implementation –

```python
# Importing libraries
import imaplib, email

user = 'USER_EMAIL_ADDRESS'
password = 'USER_PASSWORD'
```

More related articles in Python

```python
imap_url = 'imap.gmail.com'

# Function to get email content part i.e its body part
def get_body(msg):
    if msg.is_multipart():
        return get_body(msg.get_payload(0))
    else:
        return msg.get_payload(None, True)

# Function to search for a key value pair
def search(key, value, con):
    result, data = con.search(None, key, '"{}"'.format(valu
    return data

# Function to get the list of emails under this label
def get_emails(result_bytes):
    msgs = [] # all the email data are pushed inside an arr
    for num in result_bytes[0].split():
        typ, data = con.fetch(num, '(RFC822)')
        msgs.append(data)

    return msgs

# this is done to make SSL connnection with GMAIL
con = imaplib.IMAP4_SSL(imap_url)

# logging the user in
con.login(user, password)

# calling function to check for email under this label
con.select('Inbox')

 # fetching emails from this user "tu**h*****1@gmail.com"
msgs = get_emails(search('FROM', 'MY_ANOTHER_GMAIL_ADDRESS'

# Uncomment this to see what actually comes as data
# print(msgs)


# Finding the required content from our msgs
# User can make custom changes in this part to
# fetch the required content he / she needs

# printing them by the order they are displayed in your gma
for msg in msgs[::-1]:
    for sent in msg:
        if type(sent) is tuple:
```

```python
# encoding set as utf-8
content = str(sent[1], 'utf-8')
data = str(content)

# Handling errors related to unicodenecode
try:
    indexstart = data.find("ltr")
    data2 = data[indexstart + 5: len(data)]
    indexend = data2.find("</div>")

    # printtng the required content which we ne
    # to extract from our email i.e our body
    print(data2[0: indexend])

except UnicodeEncodeError as e:
    pass
```

Output:

Primary	Social 2 new LinkedIn, Twitter	Promotions	Forums	
☐ ☆ TUSHAR BAHETI	5TH MSG - All emails sent			8:17 PM
☐ ☆ TUSHAR BAHETI	4TH MSG - Hi Tushar			8:17 PM
☐ ☆ TUSHAR BAHETI	3RD MSG - This is me only			8:16 PM
☐ ☆ TUSHAR BAHETI	2nd MSG - Hello Bro			8:16 PM
☐ ☆ TUSHAR BAHETI	1ST MSG - Hello world			8:16 PM

Recommended Posts:

Fetch top 10 starred repositories of user on GitHub | Python

Send mail from your Gmail account using Python

Send mail with attachment from your Gmail account using Python

Python | Fetch Nearest Hospital locations using GoogleMaps API

Python program to fetch the indices of true values in a Boolean list

Sending Emails Using API in Flask-Mail

Take Matrix input from user in Python

md1844
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : Akanksha_Rai

Article Tags : Python   python-utility

1

0

☐ To-do ☐ Done

No votes yet.

Feedback/ Suggest Improvement   Add Notes   Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments