How to Read Emails in Python

Learn how you can use IMAP protocol to extract, parse and read emails from outlook, aol, office 365 and other email providers as well as downloading attachments using imaplib module in Python.

👤 Abdou Rockikz · 🕐 8 min read · Updated Jul 2022 · 👁 180.4K · Python Standard Library

Creating an application that can read your emails and automatically download attachments is a handy tool. In this tutorial, you will learn how to use the built-in `imaplib` module to list and read your emails in Python; we gonna need the help of **IMAP protocol**.

IMAP is an Internet standard protocol used by email clients to retrieve email messages from a mail server. Unlike the POP3 protocol, which downloads emails and deletes them from the server (and then reads them offline), with **IMAP**, the message does not remain on the local computer; it stays on the server.

If you want to read emails with Python using some sort of API instead of the standard `imaplib`, you can check the tutorial on using Gmail API, where we cover that.

Learn also: *How to Extract Google Trends Data in Python*

To get started, we don't have to install anything. All the modules used in this tutorial are the built-in ones:

```python
import imaplib
import email
from email.header import decode_header
import webbrowser
import os

# account credentials
username = "youremailaddress@provider.com"
password = "yourpassword"
# use your email provider's IMAP server, you can look for your provide
# or check this page: https://www.systoolsgroup.com/imap/
# for office 365, it's this:
imap_server = "outlook.office365.com"

def clean(text):
    # clean text for creating a folder
    return "".join(c if c.isalnum() else "_" for c in text)
```

We've imported the necessary modules and then specified the credentials of our email account. Since I'm testing this on an Office 365 account, I've used `outlook.office365.com` as the IMAP server, you check this link that contains a list of IMAP servers for the most commonly used email providers.

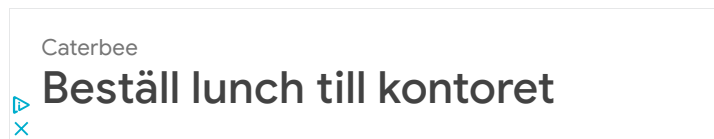**Tags**

We need the `clean()` function later to create folders without spaces and special characters.

First, we gonna need to connect to the IMAP server:

```python
# create an IMAP4 class with SSL
imap = imaplib.IMAP4_SSL(imap_server)
# authenticate
imap.login(username, password)
```

Note: *From May 30, 2022, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password. Therefore, this code won't work for Gmail accounts. If you want to interact with your Gmail account in Python, I highly encourage you to use the Gmail API tutorial instead.*

If everything went okay, then you have successfully logged in to your account. Let's start getting emails:

```python
status, messages = imap.select("INBOX")
# number of top emails to fetch
N = 3
# total number of emails
messages = int(messages[0])
```

We've used the `imap.select()` method, which selects a mailbox (Inbox, spam, etc.), we've chosen the INBOX folder. You can use the `imap.list()` method to see the available mailboxes.

`messages` variable contains a number of total messages in that folder (inbox folder) and `status` is just a message that indicates whether we received the message successfully. We then converted `messages` into an integer so we could make a `for` loop.

The `N` variable is the number of top email messages you want to retrieve; I'm gonna use 3 for now. Let's loop over each email message, extract everything we need, and finish our code:

```python
for i in range(messages, messages-N, -1):
    # fetch the email message by ID
```

New Tutorials

```python
    res, msg = imap.fetch(str(i), "(RFC822)")
    for response in msg:
        if isinstance(response, tuple):
            # parse a bytes email into a message object
            msg = email.message_from_bytes(response[1])
            # decode the email subject
            subject, encoding = decode_header(msg["Subject"])[0]
            if isinstance(subject, bytes):
                # if it's a bytes, decode to str
                subject = subject.decode(encoding)
            # decode email sender
            From, encoding = decode_header(msg.get("From"))[0]
            if isinstance(From, bytes):
                From = From.decode(encoding)
            print("Subject:", subject)
            print("From:", From)
            # if the email message is multipart
            if msg.is_multipart():
                # iterate over email parts
                for part in msg.walk():
                    # extract content type of email
                    content_type = part.get_content_type()
                    content_disposition = str(part.get("Content-Dispos
                    try:
                        # get the email body
                        body = part.get_payload(decode=True).decode()
                    except:
                        pass
                    if content_type == "text/plain" and "attachment" n
                        # print text/plain emails and skip attachments
                        print(body)
                    elif "attachment" in content_disposition:
                        # download attachment
                        filename = part.get_filename()
                        if filename:
                            folder_name = clean(subject)
                            if not os.path.isdir(folder_name):
                                # make a folder for this email (named
                                os.mkdir(folder_name)
                            filepath = os.path.join(folder_name, filen
                            # download attachment and save it
                            open(filepath, "wb").write(part.get_payloa
            else:
                # extract content type of email
                content_type = msg.get_content_type()
                # get the email body
                body = msg.get_payload(decode=True).decode()
```

```python
            if content_type == "text/plain":
                # print only text email parts
                print(body)
        if content_type == "text/html":
            # if it's HTML, create a new HTML file and open it in
            folder_name = clean(subject)
            if not os.path.isdir(folder_name):
                # make a folder for this email (named after the su
                os.mkdir(folder_name)
            filename = "index.html"
            filepath = os.path.join(folder_name, filename)
            # write the file
            open(filepath, "w").write(body)
            # open in the default browser
            webbrowser.open(filepath)
        print("="*100)
# close the connection and logout
imap.close()
imap.logout()
```

A lot to cover here. The first thing to notice is we've used `range(messages, messages-N, -1)`, which means going from the top to the bottom, the newest email messages got the highest id number, and the first email message has an ID of 1, so that's the main reason, if you want to extract the oldest email addresses, you can change it to something like `range(N)`.

Second, we used the `imap.fetch()` method, which fetches the email message by ID using the standard format specified in `RFC 822`.

After that, we parse the bytes returned by the `fetch()` method to a proper `Message` object and use the `decode_header()` function from the `email.header` module to decode the subject of the email address to human-readable Unicode.

After printing the email sender and the subject, we want to extract the body message. We look if the email message is multipart, which means it contains multiple parts. For instance, an email message can contain the `text/html` content and `text/plain` parts, which means it has the HTML and plain text versions of the

message.

It can also contain file attachments. We detect that by the `Content-Disposition` header, so we download it under a new folder created for each email message named after the subject.

The msg object, which is the `email` module's `Message` object, has many other fields to extract. In this example, we used only `From` and the `Subject`, write `msg.keys()` and see available fields to extract. You can, for instance, get the date of when the message was sent using `msg["Date"]`.

After I ran the code for my test email account, I got this output:

```
Subject: Thanks for Subscribing to our Newsletter !
From: example@domain.com
====================================================================
Subject: An email with a photo as an attachment
From: Python Code <example@domain.com>
Get the photo now!


====================================================================
Subject: A Test message with attachment
From: Python Code <example@domain.com>
There you have it!


====================================================================
```

So the code will only print `text/plain` body messages, it will create a folder for each email, which contains the attachment and the HTML version of the email. It also opens the HTML email in your default browser for each email extracted that has the HTML content.

Going to my email, I see the same emails that were printed in Python:

Awesome, I also noticed the folders created for each email:

Each folder has the HTML message (if available) and all the files attached to the email.

Conclusion

Awesome, now you can build your own email client using this recipe. For example, instead of opening each email on a new browser tab, you can build a GUI program that reads and parses HTML just like a regular browser, or maybe you want to send notifications whenever a new email is sent to you; the possibilities are endless!

A note, though, we haven't covered everything that the `imaplib` module offers. For example, you can search for emails and filter by the sender address, subject, sending date, and more using the `imap.search()` method.

Here are other Python email tutorials:

- How to Send Emails in Python.

- How to Delete Emails in Python.
- How to Use Gmail API in Python.

Here is the official documentation of modules used for this tutorial:

- imaplib — IMAP4 protocol client
- email — An email and MIME handling package
- webbrowser — Convenient Web-browser controller

Finally, if you're a beginner and want to learn Python, I suggest you take the Python For Everybody Coursera course, in which you'll learn a lot about Python. You can also check our resources and courses page to see the Python resources I recommend!

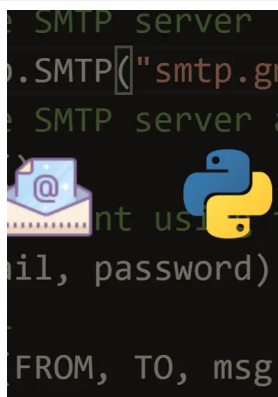**Learn also**: *How to Create a Watchdog in Python.*

Happy Coding 🖤

</> VIEW FULL CODE

Sharing is caring!

Read Also

How to Send Emails in Python

Learn how to use Python's standard library smtplib and email modules to connect to any SMTP server and send emails in Python automatically.

VISIT →
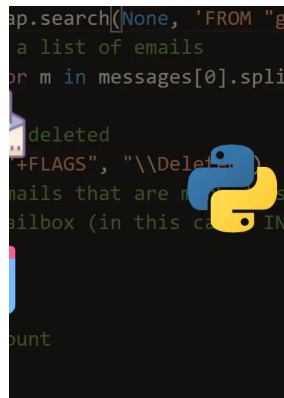
How to Delete Emails in Python

Learn how you can use IMAP protocol to delete specific mails in your email account in a selected mailbox using the built-in imaplib module in Python.

VISIT
→
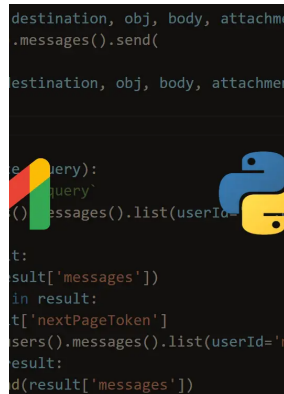
How to Use Gmail API in Python

Learn how to use Gmail API to send emails, search for emails by query, delete emails, mark emails as read or unread in Python.

VISIT
→

Comment panel

Neri Cervin 3 years ago

REPLY

Hola. Sabés de alguna manera de leer las carpetas de una cuenta IMAP, teniendo en cuenta el charset en que están generadas?

Abdou Rockikz 3 years ago

REPLY

Hi Neri, from what I understand using translator, this tutorial does exactly what you asked for, it reads the folders using IMAP and decodes it to readable string.

sanjay.dummymail@gmail.com 3 years ago

REPLY

Hello, can u say how to make the program to read only from inbox?? this program fetches mail from " promotion/social " category which i don't want to be done. your answer matters to me alot...

Abdou Rockikz 3 years ago

REPLY

Hey there,
In the tutorial, I've specified the 'INBOX' folder in imap.select() method, you can print imap.list() to see available boxes, and try all of them.
Hope this helps!

Harper B. 3 years ago   **REPLY**

Thank you so much!

Abdou Rockikz 3 years ago   **REPLY**

Your welcome Harper! Glad it helped you!

YasserKhalil 3 years ago   **REPLY**

** It seems the codes are different on GitHib from the
website. I have copied the code from GitHub.
I could solve it by using this line instead
for i in range(messages, messages-N, -1):

This line is different on GitHub (You can correct it)
https://github.com/x4nth055/pythoncode-
tutorials/blob/master/python-standard-library/reading-
email-messages/reading_emails.py

Abdou Rockikz 3 years ago   **REPLY**

Thanks Yasser for pointing out, I've edited the code
to extract the most recent top N emails, instead of
top N emails from the 4th position.

Sandy 3 years ago   **REPLY**

Hi. Thank you for sharing the tutorials. I am getting the
following error:

AttributeError: 'module' object has no attribute
'message_from_bytes'

whereas when I check my email module I did find the
function with this name. Can you please suggest what
could be the reason?

Regards

Abdou Rockikz 3 years ago   **REPLY**

Hey Sandy,
Make sure you do not have email.py in your
working directory, nor the name of the file you're
using. As Python will load it from there instead
from the Python standard library.
Please let me know if the problem persists.

A_S 3 years ago

Thank you so much! This is awesome and super super helpful!

I'm sorry to bother but a quick question that I would be grateful if you could answer: if I want to get all emails in inbox instead of just N emails, how can I do that? I tried N=len(messages) but that didn't work correctly. Thanks ever so much!

Abdou Rockikz 3 years ago

Hey there, thanks for your kind words!
You see the for loop: for i in range(messages, messages-N, -1)
If you have 300 emails in your INBOX folder, and you have set N = 10, then it'll iterate from 300 to 300-10 = 290 (the latest 10 emails).
So if you want all emails, you have to set it: for i in range(messages, 0, -1).
If that doesn't work, then use for i in range(messages, 1, -1)
Hope this helps!

TheReplenisher a year ago

the way i solved this is just under messages = int(messages[0])
i put N = messages

A_S 3 years ago

Thank you so much! That worked!!

The program works really well but sometimes it will stumble on an email and I get this error. It seems to be the only constant error:

Traceback (most recent call last):
  File "C:/Users/Ruby/PycharmProjects/email_reader/ALL_email_reader.py", line 82, in
    os.mkdir(subject)
NotADirectoryError: [WinError 267] The directory name is invalid: 'Diameter: 2.4 cm'

In this case "Diameter: 2.4cm" was an email I sent to myself and it's only contains text.

Abdou Rockikz 2 years ago

Hey there,

I think the error was because Windows does not support making directories with special characters such as colon (:), I suggest you make a function like clean_text() which removes all special characters that Windows deny on creating folders, and then you call os.mkdir(clean_text(subject)) instead of os.mkdir(subject).

I will edit the tutorial to address that soon, thanks for pointing out :)

Ruby 2 years ago

Thanks man, very helpful code.

I tried to read emails with multiple attachments in one email but the program only downloads the first attachment in an email. It doesn't work with multiple attachments per email. Any clue how I can fix that? Thank you very much for your code. Cheers.

Abdou Rockikz 2 years ago

Hey Ruby,

Thanks for pointing out, It should work with multiple attachment, as you see the for part in msg.walk(), it's iterating over all email partitions. However, I'll soon test with multiple files attached to see what's wrong, thanks again!

N.D. 2 years ago

For some reason body is not defined for me on like 65 and I can't find a reason as to why. Has anyone else had this problem?

Abdou Rockikz 2 years ago

Hey there,

What's the error raised ?

Dylan 2 years ago

How can I split the body into different variable such as split after |} and then it puts those into variable such as x, y,z?

Abdou Rockikz 2 years ago

REPLY

Hello Dylan,

You can simply use the split() method:

first_part, second_part = body.split("|}")

Ashish 2 years ago

REPLY

1. What if I have two email accounts linked to my account?
2. What if I don't know password for the shared email ID?

Abdou Rockikz 2 years ago

REPLY

Hi Ashish,

Unfortunately, imaplib requires email and password for login, If you're using Gmail, this tutorial may help you do what you need:

https://www.thepythoncode.com/article/use-gmail-api-in-python

daniel 2 years ago

REPLY

i have an error in msg = email.message_from_bytes(response[1]) (Value 'response' is unsubscriptablepylint(unsubscriptable-object))

Abdou Rockikz 2 years ago

REPLY

Hi Daniel,

I think it's just a warning from pylint which is for Python code linting, if not, please share the full traceback either here or contact here:

https://www.thepythoncode.com/contact

aliouis338@gmail.com 2 years ago

REPLY

hi, I need a help!

i am trying to recuperate emails , i use your code and it works, but not for all mails, i get errors in the emails that contient docments joints .

please what to do ?

Abdou Rockikz 2 years ago

REPLY

Hi there,

I've tried the script on multiple types of emails, including the attachments, and it worked just fine. Can you tell me what's the error raised in your case?

Brandon Wegner 2 years ago

I love it. The only issue I found was that if when it try's to get the payload if it gets an error it uses the data from the previous email. Heres a simple fix that works for me.

```
# extract content type of email
            content_type = part.get_content_type()
            content_disposition = str(part.get("Content-Disposition"))
            body="Error"
            try:
                # get the email body
                body = part.get_payload(decode=True).decode()
            except:
                try:
                    body=str(part)
                except:
                    pass
```

Abdou Rockikz 2 years ago

Hi Brandon,
Thanks for pointing out, I'll definitely try to reproduce the error and fix it with your workaround.

Michael Haraeb 2 years ago

Hello and thank you for all the info you provide.
I just wanted to know how I can move ONLY emails without attachments to a certain folder.
Please how do I make that divide?
with imaplib, python and email

Abdou Rockikz 2 years ago

Hi Michael,

You can edit this code to change the folder_name variable:

```
if content_type == "text/html":
            # if it's HTML, create a new HTML file and open it in browser
            folder_name = clean(subject) # change here
            if not os.path.isdir(folder_name):
                # make a folder for this email (named after the subject)
                os.mkdir(folder_name)
```

It is by default named after the subject of the email, you can change it whatever you want, or add something to it and it'll save the text/html contents to that folder.

Hope this helps.

Usman Gul 2 years ago

Hi,

Grateful for your code, i only have one question. I am receiving signatures in the mail most of the time. is there any way imap can separate it from the body?

Abdou Rockikz 2 years ago

Hi Usman,

You can change the folder_name variable in the attachment section in the code, as shown in the image: https://imgur.com/a/SkXaNKF

It is by default the subject name, but you can change it or add to it.

Hope this helps.

Tomos Roberts 2 years ago

Hi Abdou,

The code seems to be running correctly for me but I can't seem to find the folders it produces, any idea why? or where I should be looking?

Thanks.

Abdou Rockikz 2 years ago REPLY

Hi Tomos,

It should create folders in the current directory where you run the script, the folder names will be created based on the email subject.

Hope this helps.

Tomos Roberts a year ago REPLY

Hi Abdou,

It wasn't generating in the project directory for some reason. This was the first place I checked. Since then I've included a file path as a variable to control where my output produces and it works perfectly.

Thanks for your help, your code is very effective!

Abdou Rockikz a year ago REPLY

Awesome, glad it helped, Tomos!

BOOMSLANG 2 years ago REPLY

HEY THIS IS THE PROBLEM MY SYSTEM IS RETURNING WHAT IS REASON FOR IT:

"raceback (most recent call last):
  File "D:\PYTHON LUCIFER\imap.py", line 17, in
    imap.login(username, password)
  File "*", line 593, in login
    raise self.error(dat[-1])
imaplib.IMAP4.error: b'[AUTHENTICATIONFAILED] Invalid credentials (Failure)'"

Abdou Rockikz 2 years ago REPLY

Hello there,

There are two possibilities: Either you're using the wrong credentials, or your account is not allowing 3rd party apps to log into your account. Generally, you'll have an option to allow them.

Hope this helps!

Edbert 2 years ago REPLY

Is there a way to download the email header?

Abdou Rockikz 2 years ago   REPLY

Hi Edbert,

The `msg` object contains a lot of information
including the header, I suggest you write help(msg)
or dir(msg) to see which methods and attributes
you want to access.

Tony 2 years ago   REPLY

Hello, thank you for the awesome script!

I'm getting an error when it reaches a certain email:

os.mkdir(folder_name)

FileNotFoundError: [Errno 2] No such file or directory: ''

Thank you

Abdou Rockikz 2 years ago   REPLY

Hello Tony,

I think the email subject contains the "/" character,
you can do the replace such as:

os.mkdir(folder_name.replace("/", "-"))

This will replace the "/" with "-".

Vinoth 2 years ago   REPLY

I like to fetch all the calendar events.

imap.select("Calendar"), after this how to get the event
details please help me.

Abdou Rockikz 2 years ago   REPLY

Hello Vinoth,

In this case, you need to parse the resulting HTML.
Instead of saving the email to disk, you use
beautifulsoup library to parse the event details just
like any page on the web.

Hope this helps.

Andy a year ago   REPLY

Hi,

Is there a way to filter the pictures

Abdou Rockikz a year ago   REPLY

Hey Andy,

You can do whatever you want with the
attachments in the code block where we download
the attachment, this one:

```
elif "attachment" in content_disposition:
    # download attachment
    # your code here
```

By default we download every attachment we see,
you have the filename and you're free to do
anything you want. Hope this helps.

Andy a year ago   REPLY

Hi again,
I'm not really sure how your clean() function works, can
you please explain it?
Many Thanks!

Abdou Rockikz a year ago   REPLY

Hey Andy,

It cleans the text to be suitable for a folder name in
all operating systems, only alphanumeric and the
underscore (_) characters are allowed.

Quintin a year ago   REPLY

Hi Abdou,

Could you please assist me, I'm having issues with
embedded HTML, pictures, mails. The code works for
most mails but as soon as embedded elements are
present it returns an empty body, i tried fixing this by
doing a part.as_string but then i lose the HTML elements
of the emails. The ideal for me would be to just download
the email as an HTML file in this case but I'm struggling a
bit.

Abdou Rockikz a year ago   REPLY

Yes sure, contact me here:
https://www.thepythoncode.com/contact

Jack Kroninger a year ago

I keep getting the "imaplib.error:
b'[AUTHENTICATIONFAILED] Invalid credentials (Failure)'"
error. I receive an email from google saying there is a
suspicious app trying to access my google account, and
there is no way to override this in my google settings.

Abdou Rockikz a year ago

Hey Jack,

As mentioned in the tutorial, you have to allow less
secure app in the Google account settings here:
https://myaccount.google.com/lesssecureapps

If you can't allow them, then you need to disable 2-
step verification too, which is not suggested if it's
your main Google account.

Hope this helps.

K@m@l 12 months ago

Hi,
"To help keep your account secure, from May 30, 2022, Google no
longer supports the use of third-party apps or devices which ask you to
sign in to your Google Account using only your username and
password."
So there isn't an option to turn on to allow less secure apps.
I am getting this error by the way.
"Traceback (most recent call last):
  File "C:\Users\hp\Desktop\py4e\trial_run.py", line 18, in
    imap.login(username, password)
  File
"C:\Users\hp\AppData\Local\Programs\Python\Python39\lib\imaplib.py",
line 612, in login
    raise self.error(dat[-1])
imaplib.error: b'[AUTHENTICATIONFAILED] Invalid credentials
(Failure)'"""

Does this mean that this code won't work with g mail anymore? If it
doesn't is it possible to do the same procedure using any other method,
If so please suggest the relevant resource.

Abdou Rockikz 12 months ago

Hey there,

Unfortunately yes, it won't work with Gmail but still work on other email providers. If you want to interact with Gmail using Python, then consider using Gmail API instead, we already covered that: https://www.thepythoncode.com/article/use-gmail-api-in-python

Hope this helps!

Grant 12 months ago

nice tutorial. thank you. as of may 30 2022:

　　Less secure apps & your Google Account

　　To help keep your account secure, from May 30, 2022, Google no longer supports the use of third-party apps or devices which

　　ask you to sign in to your Google Account using only your username and password.

Abdou Rockikz 12 months ago

Hey there,

Thanks for letting me know! I invite you to use other email providers for this code to work. If you really want Gmail, then consider Gmail API, we have a tutorial on that as well:

https://www.thepythoncode.com/article/use-gmail-api-in-python

Hope this helps!

Philip Trøen 10 months ago

Hi,

I'm a bit confused as to what the difference is between the office365.com imap server and the outlook.com server. which is the right one? how do I know what server I use?

Abdou Rockikz 10 months ago

Hey Philip,

I'm not sure. But, I would guess if the URL when you're logged in, is outlook.office.com, then it's office365.com, if it's outlook.live.com, then it's outlook. Not entirely sure, but you can try both!

Rakesh Kowti 7 months ago

REPLY

raise self.error(dat[-1])

imaplib.IMAP4.error: b'LOGIN failed.'

How to resolve this problem can you please help me regarding this.

Abdou Rockikz 6 months ago

REPLY

Hey Rakesh,

If you're trying to log in on your Office365 or outlook, I think it won't work as they no longer allow IMAP logins. It happened to me as well. Check this: https://learn.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/deprecation-of-basic-authentication-exchange-online

Kit 6 months ago

REPLY

Hi Abdou,

Awesome tutorial, I'm trying to use this code but am struggling to find where I'd put a for loop to make it only check for emails with a number in its subject. Previously, I created a data frame which only has one column, a list of all the numbers I want to scan the emails for.

Would it be in the response for loop?

Many thanks

Abdou Rockikz 6 months ago

REPLY

Hey Kit,

Thanks for the kind words!

Yes, under the response for loop, under this line:

subject, encoding = decode_header(msg["Subject"])[0]

You can check whether subject has at least one number in it with this if statement:

if any(char.isdigit() for char in subject):

    // the remaining code here

Or if you mean the subject should be a number, then you can use str's isnumeric() function:

if subject.isnumeric():

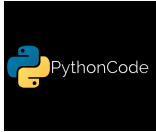   // the remaining code here

Hope this helps!

Dave 3 months ago   REPLY

Hey,

Will this code work for Gmail with the App passwords option?

Thanks in advance

Abdou Rockikz a month ago   REPLY

Hey Dave,
Unfortunately, it does not work on Gmail now.
Instead, you have to use the Gmail API:
https://www.thepythoncode.com/article/use-gmail-api-in-python

Hope this helps!

ZODIAC MA 3 months ago   REPLY

I want to know if the message go to junk folder does the imap catch it or not if not can u give me any suggestions on how to know or handle messages that goes to junk & Thank you.

Abdou Rockikz a month ago   REPLY

Hey there,
Yes, the IMAP protocol can access messages in the junk/spam folder. By default, the code provided in the tutorial reads messages from the "INBOX" folder. To access messages in the junk/spam folder, you'll need to modify the code to select that folder. Instead of "INBOX", try "Spam", "Junk" or "Junk Email"
Hope this helps,

Nancy 2 months ago   REPLY

I want to access last 5 mails which containing a word congratulation. Can please tell me?

Abdou Rockikz a month ago

Hey Nancy,

Sure! You can achieve this by using the imap.search() method with a search query that looks for the word "congratulation" in the email subject or body. Here's an example of how to modify the provided code to accomplish this:

```
# Select the INBOX folder
status, messages = imap.select("INBOX")

# Search for emails containing the word
"congratulation"
status, msg_nums = imap.search(None, 'OR
(SUBJECT "congratulation") (BODY
"congratulation")')
msg_nums = msg_nums[0].split()[-5:] # Get the
last 5 matching email IDs

# Loop over each email message
for msg_num in msg_nums:
    # fetch the email message by ID
    res, msg = imap.fetch(msg_num, "(RFC822)")
    # ... rest of the code ...
```

This code snippet selects the "INBOX" folder and then searches for emails containing the word "congratulation" in either the subject or the body. It retrieves the last 5 matching email IDs and fetches the emails accordingly. The rest of the code remains the same as in the original tutorial.

Please note that the search query is case-insensitive, so it will match "congratulation" regardless of the capitalization.

Hope this helps,

Your email address will not be published.

Enter your name

Enter Email

Write a comment...

I'm not a robot

reCAPTCHA
Privacy - Terms

☑ Subscribe for our newsletter

COMMENT

Subscribe Newsletter

Enter your Email Address

GET PYTHON TUTORIALS ➜

JOIN OUR NEWSLETTER THAT IS FOR PYTHON DEVELOPERS & ENTHUSIASTS LIKE YOU !

PRIVACY POLICY

WRITE WITH US

ADVERTISE

ABOUT

LEARNING RESOURCES

CONTACT US