


How to Send Emails with Gmail using Python

By  Scott Robinson (<https://twitter.com/ScottWRobinson>) • March 07, 2016 •
4 Comments (/how-to-send-emails-with-gmail-using-python/#disqus_thread)



There are quite a few ways to send email with Python, whether it be through a 3rd party library like with boto and SES (<http://stackabuse.com/how-to-send-an-email-with-boto-and-ses/>), or through an email protocol like SMTP. While the subject of using Python to send emails may seem like it's been done to death, there are just so many different ways to do it and so many issues that can come up. I thought it would be helpful to write up a tutorial on how to send emails with Gmail as the provider using Python.

The SMTP Protocol

This may not come as a surprise, but of course Python already has a library that lets you connect to an SMTP (https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) server, like the one Gmail uses. This library is called, predictably, `smtplib` (<https://docs.python.org/3.5/library/smtplib.html>) and comes included with Python.

SMTP (Simple Mail Transfer Protocol) is an application-level protocol (on top of TCP) used to communicate with mail servers from external services, like an email client on your phone. SMTP is a delivery protocol only, so you can't actually retrieve email with it; you can only send email, which is what we'll be focusing on in this article. If you want to retrieve email instead, then you'll want to check out the IMAP (https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol) (Internet Message Access Protocol) protocol.

I should note that many email services, like Gmail (<https://en.wikipedia.org/wiki/Gmail>), don't usually use SMTP on their internal mail servers. SMTP is usually just provided as an outward-facing interface to their service via the `smtp.gmail.com` server. This is mostly meant to be used by email clients on your phone or computer (like Outlook, Thunderbird, etc).

Opening the Connection

As already mentioned, Python conveniently comes with the `smtplib`, which handles all of the different parts of the protocol, like connecting, authenticating, validation, and of course, sending emails.

Using this library, there are a few different ways you can create a connection to your mail server. In this section, we'll focus on creating a plain, insecure connection (which should rarely, if ever, be used). This connection is unencrypted and defaults to port 25. However, the protocol for mail submission actually uses 587, which is what we'll use.

These connections are really simple to create with `smtplib`. The unencrypted version can be created with:

```
import smtplib

try:
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
except:
    print 'Something went wrong...'
```

And that's it. There really isn't much more to it than passing the server address, port, and calling `.heho()`, which identifies you to the SMTP server. Using this `server` object you can now send email over an insecure connection.

Note: You might not be ready to send emails quite yet. Gmail imposes some restrictions on SMTP connections like this. See the section "Authenticating with Gmail" below for more info.

Using a Secure Connection

When an SMTP connection is secured via TLS/SSL, it is done over port 465 and is typically called SMTPS. Needless to say, you should always use a secured connection.

There are a few different ways you can secure your SMTP connections in the `smtplib` library. The first way is to first create an insecure connection and then upgrading to TLS. This is done using the `.starttls()` method.

```
import smtplib

try:
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    # ...send emails
except:
    print 'Something went wrong...'
```

Notice that while this is very similar to the previous insecure connection we created, all that's different is that we're using the `.starttls()` method to upgrade the connection to secure.

Your other option is to create an SSL connection right from the start. In this case, you'll want to use the `.SMTP_SSL()` method instead:

```
import smtplib

try:
    server_ssl = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server_ssl.ehlo() # optional
    # ...send emails
except:
    print 'Something went wrong...'
```

Among other slight differences, we use the SMTPS port (465) right away, although you could just leave the port parameter out of this and `smtplib` will default to 465 anyway.

Now that we have our connection, let's create an email.

Creating the Email

Emails, at the very core, are just strings of text connected by newline characters. Most emails will at least have the "From", "To", "Subject" and a body fields. Here is a simple example:

```
From: you@gmail.com
To: me@gmail.com, bill@gmail.com
Subject: OMG Super Important Message
```

Hey, what's up?

- You

As you can see, each line contains a new field with its data. No binary protocol, no XML, no JSON, just line-separated strings.

A simple way to parameterize these fields is to use string formatting in Python:

```
sent_from = 'you@gmail.com'
to = ['me@gmail.com', 'bill@gmail.com']
subject = 'OMG Super Important Message'
body = 'Hey, what's up?\n\n- You'

email_text = """\
From: %s
To: %s
Subject: %s

%s
""" % (sent_from, ", ".join(to), subject, body)
```

Subscribe to our Newsletter

Get occasional tutorials, guides, and reviews in your inbox. No spam ever.

Unsubscribe at any time.

Enter your email...

Subscribe

Now all you have to do is pass the `email_text` string to `smtplib`, which we'll show in

the next section, and you're good to go.

Authenticating with Gmail

There are a few steps you need to take before you can send emails through Gmail with SMTP, and it has to do with authentication. If you're using Gmail as the provider, you'll need to tell Google to allow you to connect via SMTP, which is considered a "less secure" method.

You can't really blame Google for setting it up this way since your application (or some other 3rd party app) will need to have your plain-text password for this to work, which is definitely not ideal. It's not like the OAuth protocol where a revocable token is issued, so they have to find another way to ensure no unauthorized parties access your data.

For many other email providers you won't need to do any of the extra steps I describe here.

First, you'll want to allow less secure apps to access your account. For detailed instructions on how to do this, you should check out this page:

Allowing less secure apps to access your account (<https://support.google.com/accounts/answer/6010255>)

If you have 2-step verification enabled on your account, then you'll need to create an app-specific password for less secure apps like this. In that case, you'll need to follow the instructions here:

Sign in using App Passwords (<https://support.google.com/accounts/answer/185833>)

And finally, if you're still getting an `SMTPAuthenticationError` with an error code of

534, then you'll need to do yet another step for this to work.

Display Unlock Captcha (<https://accounts.google.com/DisplayUnlockCaptcha>)

I haven't had to do this last step for my own accounts, but I've read that it doesn't actually work right away. Apparently after you enable less secure apps, you may need to wait a few minutes before trying the 'Display Unlock Captcha' link. If you run in to this problem and find a good way around it, please let us know in the comments!

As for the actual Python code, all you need to do is call the `login` method:

```
import smtplib

gmail_user = 'you@gmail.com'
gmail_password = 'P@ssword!'

try:
    server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server.ehlo()
    server.login(gmail_user, gmail_password)
except:
    print 'Something went wrong...'
```

Sending the Email

Now that you have your SMTP connection set up and authorized your app with Google you can finally use Python to send email with Gmail.

Using the email string we constructed above, and the connected/authenticated `server` object, you need to call the `.sendmail()` method. Here is the full code, including the methods for closing the connection:

```
import smtplib

gmail_user = 'you@gmail.com'
gmail_password = 'P@ssword!'

sent_from = gmail_user
to = ['me@gmail.com', 'bill@gmail.com']
subject = 'OMG Super Important Message'
body = 'Hey, what's up?\n\n- You'

email_text = """\
From: %s
To: %s
Subject: %s

%s
""" % (sent_from, ", ".join(to), subject, body)

try:
    server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server.ehlo()
    server.login(gmail_user, gmail_password)
    server.sendmail(sent_from, to, email_text)
    server.close()

    print 'Email sent!'
except:
    print 'Something went wrong...'
```

Conclusion

Aside from the Gmail-specific authorization steps (involving less secure apps, etc), this code should work for just about any other email provider that supports SMTP access, assuming you have the correct server address and port. If you find that other providers put special restrictions on SMTP access like Google does, let us know! We'd like to include as much info as possible here.


Do you programmatically send emails with SMTP? What kind of applications do you use it for? Let us know in the comments!

Resources

- <https://docs.python.org/3.5/library/smtplib.html> (<https://docs.python.org/3.5/library/smtplib.html>)
- <http://stackoverflow.com/a/27515833/2684304> (<http://stackoverflow.com/a/27515833/2684304>)


Edit: Thanks to **cropr** (in the comments) for pointing out that Gmail imposes a number of restrictions on people sending email with SMTP. Free accounts are limited to 500 emails per day and are rate-limited to about 20 emails per second.


If you think you'll exceed these limits, then you may want to either set up your own SMTP server or use services like AWS SES or Sendgrid.


 [python \(/tag/python/\)](/tag/python/), [smtp \(/tag/smtp/\)](/tag/smtp/), [how to \(/tag/how-to/\)](/tag/how-to/), [email \(/tag/email/\)](/tag/email/)

 (<https://twitter.com>

</share?text=How%20to%20Send%20Emails%20with%20Gmail%20using%20Python&url=https://stackabuse.com/how-to-send-emails-with-gmail-using-python/>)

 (<https://www.facebook.com/sharer/sharer.php?u=https://stackabuse.com/how-to-send-emails-with-gmail-using-python/>)

 (<https://plus.google.com/share?url=https://stackabuse.com/how-to-send-emails-with-gmail-using-python/>)

 (<https://www.linkedin.com/shareArticle?mini=true%26url=https://stackabuse.com/how-to-send-emails-with-gmail-using-python/%26source=https://stackabuse.com>)



(</author/scott/>)

About Scott Robinson (</author/scott/>)

 Twitter (<https://twitter.com/ScottWRobinson>)

Subscribe to our Newsletter

Get occasional tutorials, guides, and reviews in your inbox. No spam ever. Unsubscribe at any time.

Enter your email...

Subscribe

4 Comments

StackAbuse

 Login ▾ Recommend 2 Tweet Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

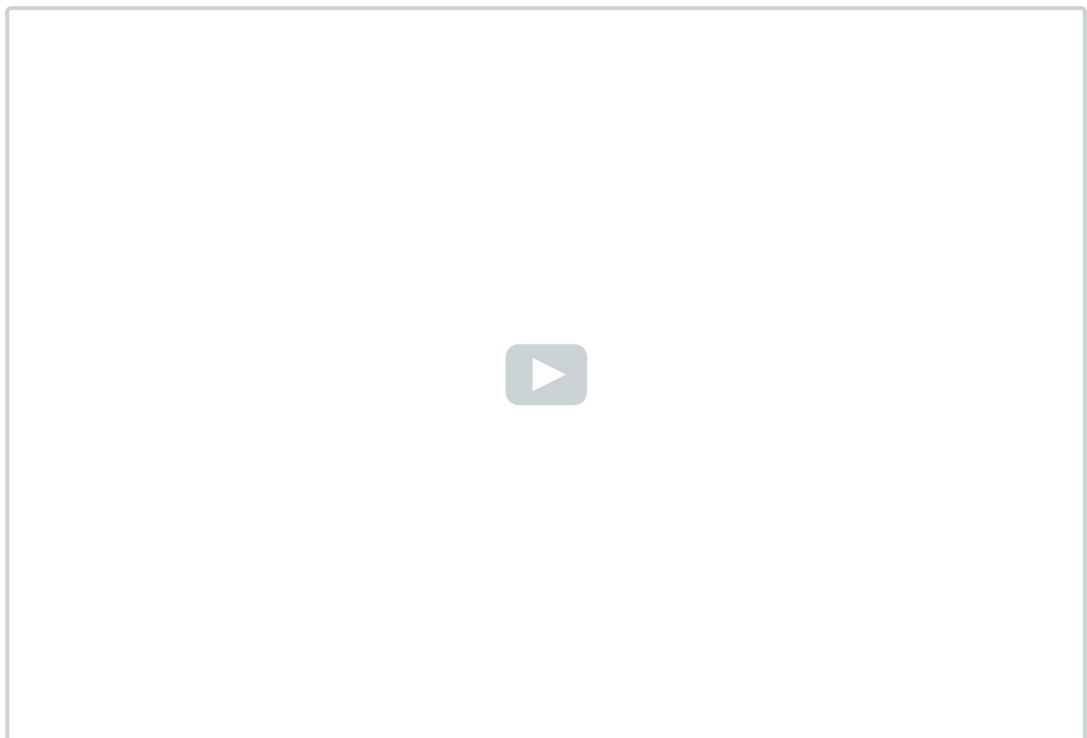
OR SIGN UP WITH DISQUS Name **Alex** • 10 months ago

The subject doesn't appear to purge through to Gmail for me when I use this exact method.

2 ^ | ▾ • Reply • Share ›

**Code&Dogs** ➔ Alex • 5 months ago

that's because you have to enable "Allow Less Secure Apps"
See details here:



^ | ▾ • Reply • Share ›

**gamesbook** • 3 months ago

Is there any reason why running the script from my machine would result in a properly formatted email; whereas running exactly the same script from a server results in a "plain text" version with no subject line?

< Previous Post : Node HTTP Servers for Static File Serving (</node-http-servers-for-static-file-serving/>)

Next Post : The Python Property Decorator > (</the-python-property-decorator/>)

Ad

Follow Us

 Twitter (<https://twitter.com/StackAbuse>)

 Facebook (<https://www.facebook.com/stackabuse>)

 RSS (<https://stackabuse.com/rss/>)

Newsletter

Subscribe to our newsletter! Get occasional tutorials, guides, and reviews in your inbox.

Subscribe

No spam ever. Unsubscribe at any time.

Ad

Want a remote job?

100% REMOTE Python Engineer

CyberCoders 14 days ago

\$100K - \$140K

([https://www.ziprecruiter.com/c/k/cybercoders-](https://www.ziprecruiter.com/c/k/cybercoders-00000000-100-remote-python-engineer-7f3bbbd4?clk=-trAfnCy0odM2oKjILN3i1DCfhA0SI_KRedzwHD2vKwrK2vzF7Stl-gZQ7V17Ayxs9YNG6t1Za3dD6_EgMgphoq8_htPEOn0CdHvxTfYBKsmSXV9brK_c9L3wAGFhLTH_dUxoSawXKmfKiGfYdDQV0Kg1TYdvU8SLHueqZznyf8K9FEVo7AwKy57X5mtllqrxlybXxgO9cDNSBRcpA_mEY-w15gIWQe88SA8s1p5dsz2qKEWic7GxebAHjyYKzYIVs5TNkw6ExHs-_sUGNKGMht1nwek57x2TIDrDYqOq71L5YrDJF0x4JhAnH8kCgjtH0saoe34kBA5RSpQ2DGnlbzbuQt3wiUVMGr8J5KTBdgMJOuq46C4dwctBTfY-Cf0Ra0UQPG0c5fDokzVGud5QE_14ZbXxemrQQtlLEIZ1iEcEL7y9nuf11paAOV1v8CHLbhBDFuj-KFk2FKrc1PF734bwt-kYoVi7LunmKYtAkp_Pvsbyd-VI4TKyxHVR15gJwFAincGj53dxbGBT7iLE-22f7mCK18NJFwJhUCr_CUay3EBsaDmXo6xfbiuKT-r-F8QtJP8vrNRYdbSLgjVY30JHCriDILDx_7Rln21MXWEjqH9SBwVB6259pVro0yoZRDxkFg2f49B822tMpF)

00000000-100-remote-python-engineer-

7f3bbbd4?clk=-

trAfnCy0odM2oKjILN3i1DCfhA0SI_KRedzwHD2vKwrK2vzF7Stl-

gZQ7V17Ayxs9YNG6t1Za3dD6_EgMgphoq8_htPEOn0CdHvxTfYBKsmSXV9brK_c9L3wAGFhLTH_dUxoSa

wXKmfKiGfYdDQV0Kg1TYdvU8SLHueqZznyf8K9FEVo7AwKy57X5mtllqrxlybXxgO9cDNSBRcpA_mEY-

w15gIWQe88SA8s1p5dsz2qKEWic7GxebAHjyYKzYIVs5TNkw6ExHs-

_sUGNKGMht1nwek57x2TIDrDYqOq71L5YrDJF0x4JhAnH8kCgjtH0saoe34kBA5RSpQ2DGnlbzbuQt3wiUV

MGr8J5KTBdgMJOuq46C4dwctBTfY-

Cf0Ra0UQPG0c5fDokzVGud5QE_14ZbXxemrQQtlLEIZ1iEcEL7y9nuf11paAOV1v8CHLbhBDFuj-

KFk2FKrc1PF734bwt-kYoVi7LunmKYtAkp_Pvsbyd-

VI4TKyxHVR15gJwFAincGj53dxbGBT7iLE-

22f7mCK18NJFwJhUCr_CUay3EBsaDmXo6xfbiuKT-

r-F8QtJP8vrNRYdbSLgjVY30JHCriDILDx_7Rln21MXWEjqH9SBwVB6259pVro0yoZRDxkFg2f49B822tMpF

Python Developer - Remote - Django/AppEngine

CyberCoders 28 days ago

\$97K - \$150K

(<https://www.ziprecruiter.com/clk/cybercoders-00000000-python-developer-remote-django-appengine->

921a14a9?clk=WjV5JN40XeVraDg8UGTugkxtsxRr8A2kij3P9oX570fiT51m1GBQdbYpJg7StUGjnYhZgDz4IlpXQdDj5yACLPjdT_9U0BFnb9I7GHmBIL-faSlx3C2G2QDFySeZtt42scZyEJtiDWmG_ulSCmFksgicG5hM0Br-L7EgO273E6xA5fX9zyuhsOZ3bJJmphGiAN8alxIHPhf7LFYtyQaHxp6-_SXu5_8ZcX11hwo5ppnZZWQrxCyVePi4eauoePg4HdRfvK765hzXdzbWFhK0zW7Z0ghlh_eoayFtcgRL1N8kC9ilx2BE-Y3EI-YAZ4aymC2dtWeDKKBgfhQZmfS-Si1eNjLSasetAoOW61HcqghCtT6exSQ2ioDq5g_Ty52bvVey7abM3UsDPoVs9fZMdpQCN4f6F6RZI7ck0Pqi0CThrdFilzCy_oHjOnh_rNG7sEECH0qD4HVZJUTzurJZUw4hJoffBs4xJw-ks4p6kzC8Ew1p08MhT8RSh7OXL57sTLYk-ueAlpX6oEUz0k5gaAOppj4Q2V5Yz72TY3sBnEvr6kbtjKhDSw.a8201b7799cda0667a7a1e58ad00e0ae)

Sr Python Developer Remote

CriticalRiver Inc 13 days ago

\$140K - \$150K

(https://www.ziprecruiter.com/clk/criticalriver-inc-00000000-sr-python-developer-remote-3348ca11?clk=JE5cQWbvglV0uJwRoeUeX0KAF1n19xK9TVVYg-sLSfydIDfLiZIGbu_x_-vh21pafD7CDGAmP8DKtnMVbE1iS9hcy7vBfGhviaK4uMat2doKjEnEG0aC4SRYmkSj27sEbUvOqRPDKJB5JoTzPomS3Q1fg49d1zFXJI-FAIj9Hub283SzeC6h6epY-o-RksoUQ_xbvVpWy6l_iLsxuiVQCUKaaiH_G5zuZXd5sNJ7tNwmMSUJsJkHQHkLuTlv0NIDDOMcBNn4lapmS8nqCnxLbOLOWC0l1bQ4JlbqgZT46URe49bX3faT6bcnfD34US3nMIDA_0jqw6zj4_LA1dN7tZ8f3rte3zLltvHUO-6JfdWdUbGyhiZsf3n_T7E1x7aenLA-KCqP5X9gd50GLlrVzZZuG41uXwZ6ImCv8PBe_Kbxn0YdXWrrMjJ89h2I5svEx2qwDtM8fS0EDy7EZaB17ffA

➔ More jobs (<https://hireremote.io>)

Interviewing for a job?

(<http://stackabu.se/daily-coding-problem>)

- Improve your skills by solving one coding problem every day
- Get the solutions the next morning via email
- Practice on **actual problems** asked by top

companies, like:

`</>` Daily Coding Problem (<http://stackabu.se/daily-coding-problem>)

Recent Posts

Python: Print without
Newline
(<https://stackabuse.com/python-print-without-newline/>)
July 8, 2019

The Python Assert
Statement
(<https://stackabuse.com/the-python-assert-statement/>)
July 8, 2019

The Python Help System
(<https://stackabuse.com/the-python-help-system/>)
July 5, 2019

Tags

[python \(/tag/python\)](/tag/python)

[nlp \(/tag/nlp\)](/tag/nlp)

[machine learning \(/tag/machine-learning\)](/tag/machine-learning)

[tensorflow \(/tag/tensorflow\)](/tag/tensorflow)

[angular \(/tag/angular\)](/tag/angular)

[javascript \(/tag/javascript\)](/tag/javascript)

[java \(/tag/java\)](/tag/java)

[spring \(/tag/spring\)](/tag/spring)

[phaser \(/tag/phaser\)](/tag/phaser)

[game development \(/tag/game-development\)](/tag/game-development)


[how to \(/tag/how-to\)](/tag/how-to)


[amqp \(/tag/amqp\)](/tag/amqp)


[git \(/tag/git\)](/tag/git)

[concurrency \(/tag/concurrency\)](/tag/concurrency)

Follow Us

 Twitter
(<https://twitter.com/StackAbuse>)

 Facebook
(<https://www.facebook.com/stackabuse>)

 RSS
(<https://stackabuse.com/rss/>)

[Disclosure \(/disclosure\)](#) • [Privacy Policy \(/privacy-policy\)](#) • [Terms of Service \(/terms-of-service\)](#)