

Mass Mail Dispatcher

GAUTHAM PAI

# Introduction

This project is required to send a message to multiple email addresses, using Python. For, this Secure Mail Transfer Protocol (SMTP) must be used. This is the protocol that most email providers (such as Gmail, Outlook and Yahoo!) use to send emails securely. In SMTP, there are two types of authentications that can be used: Secure Socket Layer (SSL) and Transport Layer Security (TLS). The type of authentication is decided by the mail provider. Common mail providers, such as Gmail and Yahoo! Use both kinds of authentication. Depending on the type of authentication, the data must be sent to different ports (465 for SSL and 587 for TLS).

The project must also be able to tell between correct and incorrect email addresses. For this, a basic regex string can be used ([^.+@.+\..+$](mailto:%5e.+@.+\..+$)). This regex string provides the basics of an email address. If an email address does not match this regex string, it means that the email address is invalid.

## yagmail API

Because of the above requirements, an SMTP server address (pertaining to the email provider) and the type of authentication is required. A basic user may not know this information, nor they might know about the meaning of those fields. Therefore, the yagmail API can be used. This API automatically determines the SMTP server address (such as smtp.gmail.com for Google and smtl.mail.yahoo.com for Yahoo! Mail) and the type of authentication.

# Explanation of the code

The code is as follows:

import csv

import re

import yagmail

from email.mime.text import MIMEText

# Obtain details from user

msg = MIMEText(input('Please enter your message: '))

msg['Subject'] = input('Please enter your subject: ')

sender = input('Please enter your email address: ')

msg['From'] = sender

password = input('Please enter your password: ')

# Start connection to server

server = yagmail.SMTP(sender, password)

# Obtain the CSV file location

email\_data = csv.reader(open(input("Please enter the location of the CSV file: "), 'r'))

# Send emails

email\_pattern = re.compile("^.+@.+\..+$")

for row in email\_data:

if email\_pattern.search(row[0]):

del msg['To']

msg['To'] = row[0]

print (row[0] + '\t Valid Address')

try:

server.send([row[0]], msg['Subject'], msg.as\_string())

except yagmail.YagAddressError:

print("An error occurred.")

else:

print(row[0] + '\t Invalid Address')

The code first imports the following packages:

* csv – This package is used to read and parse the csv file containing the email addresses
* re – This package is used to implement the regex such that it can recognise invalid email addresses.
* yagmail - This package is used to communicate with the email provider’s SMTP server and send the email
* MIMEText – This package is used to contain the content of the email.

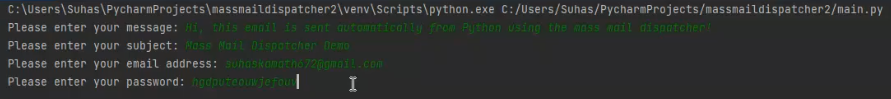
Then, the program obtains the following details from the user:

* Message – The message that needs to be in the email.
* Subject – The subject of the email.
* Sender – The sender of the email (the email address from which the email must be sent)
* Sender’s password – The password of the email address from the email must be sent.

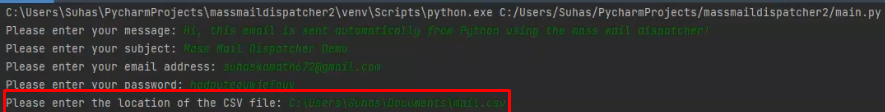
Using the user’s credentials, the program contacts the SMTP server of the email provider. Then, the user is asked for the location of the csv file containing the email addresses. Finally, the program sends an email to all the valid emails. The program lists all the emails, and whether they are valid or invalid.

# Execution of the code – Step by Step

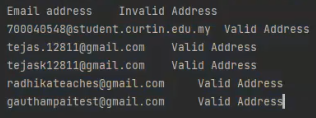
1. The user is asked for the message, subject, sender email and sender password. The details must be entered and the new line key must be pressed.



1. The user is then required to enter the location of the csv file containing the email addresses:



1. The emails are sent. The email addresses are listed, followed by whether they are valid or invalid.



Some email servers, such as Gmail do not allow the user to use the actual password for their account. Therefore, the use may first have to create an “App Password”. This can be created using the following steps:

1. Go into the account management page of the email provider (such as myaccount.google.com for Gmail)
2. Go to the “Security” or “Security and Privacy” tab.
3. Search for “App Passwords”
4. Create an app password, and note it down. Please note that the app password is displayed only once, for security reasons.
5. Use the aforementioned app password rather than your actual password when the program asks for your password.