

# Job Application Filter

By Joshua Paul

## Objective:

Create a Python script that can parse through a gmail inbox and locate emails associated with job applications.

## Libraries:

Imaplib - connect to and search through email service

Email - handle obtain email contents

os - allow for creation and editing of files

Pandas - handle data formulation after obtaining emails

### Step 1:

Create yaml file to hold username and IPOP password.

Use imaplib to SSL into Gmail service

```
with open("credentials.yaml") as f:
    content = f.read()

# from credentials.yaml import user name and password
my_credentials = yaml.load(content, Loader=yaml.FullLoader)

#Load the user name and passwd from yaml file
user, password = my_credentials["user"], my_credentials["password"]

#URL for IMAP connection
imap_url = 'imap.gmail.com'

# Connection with GMAIL using SSL
my_mail = imaplib.IMAP4_SSL(imap_url)

# Log in using your credentials
my_mail.login(user, password)

# Select the Inbox to fetch messages
my_mail.select('inbox')
```

### Step 2:

Search for emails with “application, “screen”, and “interview” in them.

Place these emails into an array and remove the duplicates

```
#Define Key and Value for email search
#For other keys check out imap_keys.txt document
_, data1 = my_mail.search(None, "BODY","application")
print("Found ", len(data1[0].split()), " emails containing application")

_, data2 = my_mail.search(None, "BODY","screen")
print("Found ", len(data2[0].split()), " emails containing screen")

_, data3 = my_mail.search(None, "SUBJECT","interview")
_, data4 = my_mail.search(None, "BODY","interview")
print("Found ", len(data3[0].split()) + len(data4[0].split()), " emails")

dataai = data1 + data2 + data3 + data4
dataai = dataai[0].split()
dataaf = list(set(dataai)) #remove duplicate emails
print("Removing duplicate emails and extracting the rest")

mail_id_list = dataaf #IDs of all emails that we want to fetch
msgs = [] # empty list to capture all messages
```

### Step 3:

Extract data from emails, remove certain words from sender ID, check if Excel already exists and delete it if it does

```
#Iterate through messages and extract data into the msgs list
for num in mail_id_list:
    typ, data = my_mail.fetch(num, '(RFC822)') #RFC822 returns wh
    msgs.append(data)

print("retrieved ", len(mail_id_list), " emails")

#words to remove from sender
remove = ["com", "online", "apply", "workday", "career", "talent", "da
        , "my", "linkedin", "indeed", "no", "reply", "talent", "recrui
        "Workday", "mail"]

#check if excel sheet already exists
path = 'C:\\Users\\jcpau\\OneDrive\\Desktop\\coding stuff\\Email\\
exists = os.path.isfile(path)

#create the excel sheet
if(exists == True):
    print("excel found, creating new sheet")
    os.remove("applications1.xlsx")
else:
    print("Creating excel sheet")

sender= []
application = []
subject1 = []
```

### Step 4:

Iterate through all extracted emails, remove special characters and save sender ID and Subject.  
Parse through body of email and label if the email is for an application, phone screening, or interview

```
for msg in msgs[::-1]:
    step = 'application'
    body = "none"
    subject = "none"
    for response_part in msg:
        if type(response_part) is tuple:
            my_msg=email.message_from_bytes((response_part[1]))
            #Remove special characters from subject and sender
            subject = ''.join(letter for letter in my_msg["subject"] if letter.isalnum())
            f = ''.join(letter for letter in my_msg['from'] if letter.isalnum())
            for i in remove:
                f = f.replace(i, '')
            #clean up the body portion of email
            for part in my_msg.walk():
                #print(part.get_content_type())
                if part.get_content_type() == 'text/plain':
                    body = part.get_payload()
                    #categorize what step of application each email is
                    if "interview" in subject or "Interview" in subject:
                        step = "interview"
                    elif "screen" in body or "screen" in subject:
                        step = "screen"

            sender.append(f)
            application.append(step)
            subject1.append(subject)
```

## Final step:

Add data to a Pandas Dataframe for easy excel formatting

Emails with interview in the subject means you made it to the interview, screen in the body or subject would be a phone screening, and the rest are applications

## Output

Show user how many of the retrieved emails are applications, a phone screening, or interview.

Export data into an excel sheet in case user wants to keep track of applications that way.

```
Found 817 emails containing application
Found 68 emails containing screen
Found 115 emails containing interview
Removing duplicate emails and extracting the rest
retrieved 817 emails
excel found, creating new sheet
Applications: 815
screenings: 2
interviews: 1
--- 115.57093477249146 seconds ---
```

sender	step	subject
dot	application	RNSupervisorRosecransCareCenter
LinkedIn	application	Joshua your application was sent to Conch Technologies Inc
OMNIVISION Recruiting Team	application	Your application for Digital Design Engineer at OMNIVISION
humanresourcesintelliswift	application	EEO Self Identification Form
US Patent Trademark Office	application	Interview with US Patent Trademark Office
LinkedIn	application	Joshua your application was sent to Insight Global
Workable candidates	application	Thanks for applying to Perceptronics Solutions Inc
dot	application	Software Engineer ARVR MR Leadership Meta
SAIC Careers	application	Thank you for your interest in SAIC
us staffing office opm gov	application	Notice of Results Patent Examiner Computer Engineer PATSE
LinkedIn	application	Joshua your application was sent to Bitus Labs
No Reply sterling app	application	On behalf of Space Exploration Technologies Background Scr
Moog Human Resources	application	Moog Application Status FPGA Design Engineer Hybrid 98024
us greenhouse io	application	Thank you for applying to Keyfactor Inc
us greenhouse io	application	Astranis Thank You for Launching Your Application into Orbit
LinkedIn	application	Joshua your application was sent to Consult Net Technology S
Arion sars services	application	We Received Your Application