import imaplib

import os

import email

from base64 import b64decode

import sys

import csv

import json

class GMAIL\_EXTRACTOR():

def helloWorld(self):

print("\nWelcome to Gmail Extractor.")

def initializeVariables(self):

self.user = ""

self.pwd = ""

self.mail = object

self.mailbox = ""

self.mailCount = 0

self.destFolder = ""

self.data = [ ]

self.ids = [ ]

self.idsList = [ ]

def getLogin(self):

print("\nPlease enter your gmail login details below.")

self.user =input("Email: ")

self.pwd=input("Password: ")

def attemptLogin(self):

self.mail = imaplib.IMAP4\_SSL("imap.gmail.com", 993)

if self.mail.login(self.user, self.pwd):

print("\nLogin SUCCESSFUL")

self.destFolder = input("\nPlease choose a destination folder in the form of /Users/username/dest/ (do not forget trailing slash!): ")

if not self.destFolder.endswith("/"): self.destFolder+="/"

return True

else:

print("\nLogon FAILED")

return False

def selectMailbox(self):

self.mailbox=input("\nPlease type the name of the mailbox you want to extract, e.g. Inbox: ")

bin\_count=self.mail.select(self.mailbox)[1]

self.mailCount=int(bin\_count[0].decode("utf-8"))

return True if self.mailCount > 0 else False

def searchThroughMailbox(self):

type,self.data=self.mail.search(None, "ALL")

self.ids=self.data[0]

self.idsList=self.ids.split()

def checkIfUsersWantsToContinue(self):

print("\nWe have found" + str(self.mailCount) + "emails in the mailbox" + self.mailbox + ".")

return True if input("Do you wish to continue extracting all the emails into "+self.destFolder+"?(y/N)").lower().strip()[:1]=="y" else False

def parseEmails(self):

csvOutput = {}

for anEmail in self.data[0].split():

type, self.data = self.mail.fetch(anEmail, '(UID RFC822)')

raw = self.data[0][1]

try:

raw\_str = raw.decode("utf-8")

except UnicodeDecodeError:

try:

raw\_str = raw.decode("ISO-8859-1") # ANSI support

except UnicodeDecodeError:

try:

raw\_str = raw.decode("ascii") # ASCII ?

except UnicodeDecodeError:

pass

msg = email.message\_from\_string(raw\_str)

csvOutput['From'] = msg['From']

csvOutput['To']= msg['To']

csvOutput['Date']= msg['Date']

csvOutput['Subject']= msg['Subject']

raw = self.data[0][0]

raw\_str = raw.decode("utf-8")

uid = raw\_str.split()[2]

# Body #

if msg.is\_multipart():

for part in msg.walk():

partType = part.get\_content\_type()

## Get Body ##

if partType == "text/plain" and "attachment" not in part:

csvOutput['body'] = part.get\_payload()

## Get Attachments ##

if part.get('Content-Disposition') is None:

attchName = part.get\_filename()

if bool(attchName):

attchFilePath = str(self.destFolder)+str(uid)+str("/")+str(attchName)

os.makedirs(os.path.dirname(attchFilePath), exist\_ok=True)

with open(attchFilePath, "wb") as f:

f.write(part.get\_payload(decode=True))

else:

csvOutput['body'] = msg.get\_payload(decode=True).decode("utf-8") # Non-multipart email, perhaps no attachments or just text.

outputDump = json.dumps(csvOutput)

emailInfoFilePath = str(self.destFolder)+str(uid)+str("/")+str(uid)+str(".csv")

os.makedirs(os.path.dirname(emailInfoFilePath), exist\_ok=True)

with open(emailInfoFilePath, "w") as f:

f.write(outputDump)

def \_\_init\_\_(self):

self.initializeVariables()

self.helloWorld()

self.getLogin()

if self.attemptLogin():

not self.selectMailbox() and sys.exit()

else:

sys.exit()

not self.checkIfUsersWantsToContinue() and sys.exit()

self.searchThroughMailbox()

self.parseEmails()

if \_\_name\_\_ == "\_\_main\_\_":

run = GMAIL\_EXTRACTOR()

**App script google sheets**

var ui = SpreadsheetApp.getUi();

function onOpen(e){

  ui.createMenu("Export Gmail").addItem("Get Emails by Label", "getGmailEmails").addToUi();

}

function getGmailEmails(){

  var input = ui.prompt('Label Name', 'Enter the label name that is assigned to your emails:', Browser.Buttons.OK\_CANCEL);

  if (input.getSelectedButton() == ui.Button.CANCEL){

    return;

  }

  var label = GmailApp.getUserLabelByName(input.getResponseText());

  var threads = label.getThreads();

  for(var i = threads.length - 1; i >=0; i--){

    var messages = threads[i].getMessages();

    for (var j = 0; j <messages.length; j++){

      var message = messages[j];

      if (message.isUnread()){

        extractDetails(message);

        GmailApp.markMessageRead(message);

      }

    }

    threads[i].removeLabel(label);

  }

}

function extractDetails(message){

  var dateTime = message.getDate();

  var subjectText = message.getSubject();

  var senderDetails = message.getFrom();

  var receiverDetails = message.getTo();

  var bodyContents = message.getPlainBody();

  var activeSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  activeSheet.appendRow([dateTime, senderDetails, receiverDetails, subjectText, bodyContents]);

}

**APP SCRIPT-2 SHEET NAME- GMAILS EMAILS**

var ui = SpreadsheetApp.getUi();

function onOpen(e){

  ui.createMenu("Export Gmail").addItem("Get Emails by Label", "getGmailEmails").addToUi();

}

function getGmailEmails(){

  var input = ui.prompt('Label Name', 'Enter the label name that is assigned to your emails:', Browser.Buttons.OK\_CANCEL);

  if (input.getSelectedButton() == ui.Button.CANCEL){

    return;

  }

  var label = GmailApp.getUserLabelByName(input.getResponseText());

  var threads = label.getThreads();

  for(var i = threads.length - 1; i >=0; i--){

    var messages = threads[i].getMessages();

    for (var j = 0; j <messages.length; j++){

      var message = messages[j];

      if (message.isUnread()){

        extractDetails(message);

        GmailApp.markMessageRead(message);

      }

    }

  }

}

function extractDetails(message){

  var input = ui.prompt('Label Name', 'Enter the label name that is assigned to your emails:', Browser.Buttons.OK\_CANCEL);

  if (input.getSelectedButton() == ui.Button.CANCEL){

    return;

  }

  var ss = SpreadsheetApp.getActiveSpreadsheet();

  var sheet = ss.getSheetByName("Sheet1");

  var label = GmailApp.getUserLabelByName(input.getResponseText());

  var threads = label.getThreads();

  for (var i= 0; i< threads.length;i++) {

    var messages = threads[i].getMessages();

     for (j = 0; j<messages.length; j++) {

  var dateTime = messages[j].getDate();

  var subjectText = messages[j].getSubject();

  var senderDetails = messages[j].getFrom();

  var receiverDetails = messages[j].getTo();

  var bodyContents = messages[j].getPlainBody();

  var activeSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  activeSheet.appendRow([dateTime, senderDetails, receiverDetails,  bodyContents]);

  Logger.log(bodyContents)

  getGmailEmails()

     }

  }

}