Week 11 : Email Automation

- ตั้งชื่อ Project ด้วย week11_รหัสนักศึกษา เช่น week11_62070001
- ในแต่ละข้อ สร้าง sequence flow ด้วย เลขที่ข้อ_รหัสนักศึกษา เช่น 11.1_62070001
- บันทึก flow แต่ละข้อให้เห็นชัดเจนพร้อมกับ run ผลลัพธ์ แสดงผลลัพธ์ในแต่ละข้อ และบันทึกคลิป ตามรูปแบบ เช่น week11_62070001.mp4 (ใช้โปรแกรมบันทึกหน้าจอ)
- ส่งงานทุกข้อเป็น 1 Project รวมใน 1 folder ที่มี input, output ทั้งหมดในโปรเจคนั้นๆ ตั้งชื่อไฟล์ week11_รหัสนักศึกษา.zip เช่น week11_62070001.zip

Prerequisite:

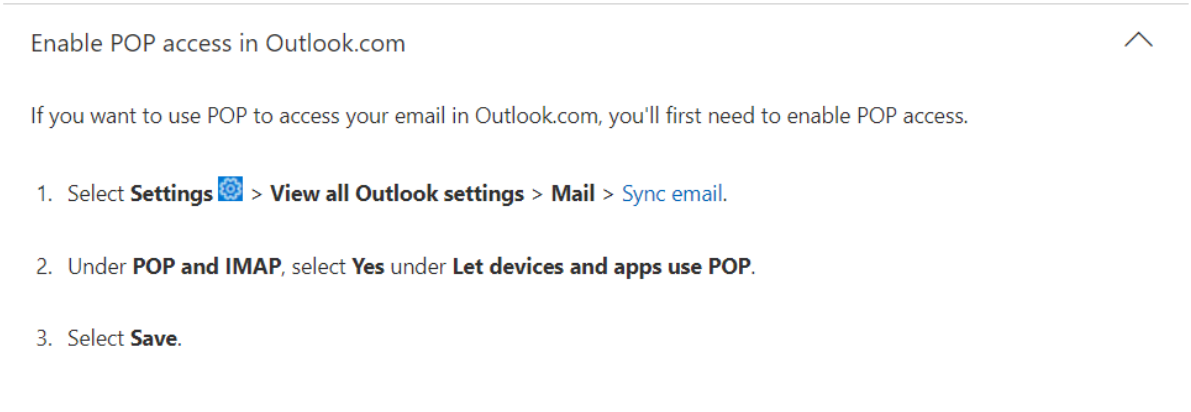
1. Update UiPath.Mail.Activities Package

2. Enabling Gmail for Email Activities

https://docs.uipath.com/studio/v2019/docs/enabling-gmail-for-email-activities

3. If using Gmail username and password, then allow Google Account for less secure apps and devices. Turn on Less Secure App at https://myaccount.google.com/lesssecureapps

4. For Outlook.com, Hotmail.com



POP Setting

Server name: outlook.office365.com

Port: 995

Encryption method: TLS

IMAP Setting

Server name: outlook.office365.com

Port: 993

Encryption method: TLS

SecureConnection: Auto

SMTP Setting

Server name: smtp.office365.com

Port: 587

Encryption method: STARTTLS

****

Gmail Setting

SMTP Server: smtp.gmail.com

Port: If SSL is enabled use Port 465.

If TLS is enabled use Port 587.

IMAP Setting

Server name: imap.gmail.com

Port: 993

Exercise 11.1 : Extract attachments from the emails

Objective: Build a workflow that extracts attachments from the emails containing the word “Resume” in its subject.

- Set up IMAP configuration to access the email.
- Loop through each email to identify subjects containing the word “Resume” or “resume”.
- Download the attachments from the identified emails in a folder.

Prerequisite: Send 3 dummy emails to the email account that you will use for practice. Emails must contain any file attachments and the word “Resume” in their subject. (example: resume 1, Resume_submission, resume:test)

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **“Sequence”** activity.

✓ **“Get IMAP Mail Messages”** activity.

✓ **“For Each**” activity.

✓ **“If”** activity.

✓ **“Save Attachments”** activity.

Algorithm:

Step 1: START

Step 2: Use Get IMAP Mail Messages activity to retrieve email data.

- Mailfolder: “Inbox”
- Port: 993
- Server: setting as same as receiver’s email account
- Email: receiver’s email account
- Password: receiver’s email password
- Set Messages in Output property as a new variable “mailResumes”

Step 3: Use For Each activity to iterate through retrieved each email.

- Set TypeArgument Property to “System.Net.Mail.MailMessage”
- Set Values Property to “mailResumes”

Step 4: Use If activity within the For Each activity and check for word “Resume” or “resume” in their Subject

- Checks if the email “Subject” contains resume or Resume. If yes, it saves the attachment.
- Condition > item.Subject.Contains("resume") OR item.Subject.Contains("Resume")

Step 5: Use Save Attachment activity within the If activity to store attachments that contains word “resume” in the Subject.

- FolderPath : in the folder named “attach” under project folder

Step 6: STOP

Outcome:

There are three files from the emails in the destination path.

Remark: If found following error message, just turn off anti-virus

Get IMAP Mail Messages: An error occurred while attempting to establish an SSL or TLS connection.

“One possibility is that you are trying to connect to a port which does not support SSL/TLS.

The other possibility is that the SSL certificate presented by the server is not trusted by the system for one or more of the following reasons:

1. The server is using a self-signed certificate which cannot be verified.

2. The local system is missing a Root or Intermediate certificate needed to verify the server's certificate.

3. The certificate presented by the server is expired or invalid.

See https://github.com/jstedfast/MailKit/blob/master/FAQ.md#InvalidSslCertificate for possible solutions.

Exercise 11.2 : Extract information and store in an CSV file

Objective: Demonstrate how to extract From, Subject, and Body of emails using Get IMAP Mail Messages activity and store in a CSV file.

- Set up IMAP configuration to access the email
- Loop through each email (top 10 of emails) and extract From, Subject, and Body
- Use Workbook to store extracted data in an excel file in three different columns with header names as From, Subject, and Body

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Build Data Table"** activity.

✓ **"Get IMAP Mail Messages"** activity.

✓ **"For Each**" activity.

✓ **"Assign"** activity.

✓ **"Add Data Row"** activity.

✓ **"Write CSV"** activity.

Algorithm:

Step 1: START

Step 2: Insert a **Build Data Table** in the designer panel and create three columns with headers as **Subject** (String), **From** (String), and **Body** (String).

- Go to the Properties panel of this activity and in DataTable property, hit Ctrl+K and enter a new variable called **getMails**.

Step 3: Insert **Get IMAP Mail Messages** activity in the designer panel.

- In the MailFolder property, enter "Inbox".
- In the Port property, enter 993.
- In Server property enter "imap.gmail.com". In the Email property, enter your email address in double quotes and in the Password property, enter your password in double quotes.

- Ensure that the OnlyUnreadMessages property is unchecked. It will fetch all emails regardless of their Read status.
- In the Top property, enter 5 (meaning top 5 of emails).
- In the Messages property, hit Ctrl+K and enter a variable **newMessages**.

Step 4: Insert a **For Each activity** below the Get IMAP Mail Messages activity. Replace the text "item" from the first text box with "**mail**", and in the adjacent text box enter the variable **newMessages**.

- Go to the Properties panel of the For Each activity and click the drop down of TypeArgument property and select "**Browse for Types**".
- In the pop up window, browse for **MailMessage** type. You can notice that there are two results shown. First is under **System.Net.Mail** and the second is under System.Web.Mail. Pick the first one.

Step 5: Insert **Assign activity** in the Body section of the For Each activity. In the first text box, hit Ctrl+K and enter a new variable called **subject**. In the adjacent box enter the expression: **mail.Subject**.

Step 6: Insert **second Assign activity** below the previous Assign activity. In the first text box, hit Ctrl+K and enter a new variable called **from**. In the adjacent box enter the expression: **mail.from.ToString**.

Step 7: Insert **third Assign activity** below the second Assign activity. In the first text box, hit Ctrl+K and enter a new variable called **body**. In the adjacent box enter the expression: **mail.Body**

Step 8: Insert **Add Data Row** activity below the third Assign activity. Go its properties panel, and in ArrayRow property, enter the expression: **{subject,from,body}**. Also, in the DataTable property, enter the variable **getMails**.

Step 9: Insert a **Write CSV** activity after For Each activity. In FilePath, put **gmailMails.csv** and in datatable text box enter the data table variable **getMails**.

Step 10: Save and run the workflow.

Step 11: Open the CSV file gmailMails.csv

Outcome:

All the emails are stored in separate columns under Subject, From, and Body in **gmailMails.csv**

Exercise 11.3 : Extract information and store in an excel file

Objective: Demonstrate how to extract and store From, Subject, and Body of emails using Get IMAP Mail Messages activity and store in an excel file

- Use Get IMAP Mail Messages activity to identify Outlook mail folder
- Loop through each email and extract From, Subject, and Body
- Use Excel Scope to store extracted data in an excel file in three different columns with header names as From, Subject, and Body

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"** activity.

✓ **"Build Data Table"** activity.

✓ **"Get IMAP Mail Messages"** activity.

✓ **"For Each**" activity.

✓ **"Assign"** activity.

✓ **"Add Data Row"** activity.

✓ **"Excel Scope"** activity.

✓ **"Write Range"** activity.

Algorithm:

Step 1: START

Step 2: Before starting this activity, ensure that you have configured Microsoft Outlook on your computer and it is in working state.

Step 3: Now, insert a **Build Data Table** in the designer panel. Click on the Data Table button within the activity. In the pop up window, create three columns with headers as Subject with Data Type as String, From with Data Type as String, and Body with Data Type as String. Go to the Properties panel of this activity and in DataTable property, hit Ctrl+K and enter a new variable called **getMails**.

Step 4: Insert **Get IMAP Mail Messages** activity in the designer panel.

- Go to its Properties panel. In the Account property, specify the email account in double quotes from where you want to fetch the mails.

- In the MailFolder property, enter the folder name in double quotes from where you want to fetch mails.
- Ensure that the OnlyUnreadMessages property is unchecked.
- It will fetch all emails regardless of their Read status. In the Top property, enter 10.
- In the Messages property, hit Ctrl+K and enter a variable **newMessages**.

Step 5: Insert a **For Each** activity below the Get Outlook Mail Messages activity. Replace the text "item" from the first text box with "**mail**", and in the adjacent text box enter the variable **newMessages**.

- Go to the Properties panel of the For Each activity and click the drop down of TypeArgument property and select "**Browse for Types**".
- In the pop up window, browse for MailMessage type. You can notice that there are two results shown. First is under System.Net.Mail and the second is under System.Web.Mail. Pick the first one.

Step 6: Insert an **Assign** activity in the Body section of the For Each activity. In the first text box, hit Ctrl+K and enter a new variable called **subject**. In the adjacent box enter the expression: **mail.Subject.**

Step 7: Insert second **Assign** activity below the previous Assign activity. In the first text box, hit Ctrl+K and enter a new variable called **from**. In the adjacent box enter the expression: **mail.from.ToString.**

Step 8: Insert third **Assign** activity below the second Assign activity. In the first text box, hit Ctrl+K and enter a new variable called **body**. In the adjacent box enter the expression: **mail.Body**

Step 9: Insert **Add Data Row** activity below the third Assign activity. Go its properties panel, and in **ArrayRow** property, enter the expression: {subject,from,body}. Also, in the DataTable property, enter the variable **getMails**.

Step 10: Insert an **Excel Application Scope** activity below the For Each activity.

- In the Workbook path text box, enter the text "**outlookMails.xlsx**".
- Insert a **Write Range** activity within the Do container. Make sure that you pick the Write Range activity that is under Excel category from the Activities panel. In the Data table text box of the Write Range activity, enter **getMails**.

Step 10: Save and run the workflow.

Step 14: Open the excel file **outlookMails.xlsx**

Outcome:

All the emails are stored in separate columns under Subject, From, and Body in excel file **outlookMails.xlsx**

Exercise 11.4: Send email with text from two files

Objective: Demonstrate how to extract text from native pdf and scanned pdf and send them via email using Send SMTP Mail Messages activity.

- Reads the 6th page of "UiPathOrchestratorAzureInstallationGuide2016.1.pdf"
- Reads the 2nd page of "ScannedDoc.pdf"
- Sends an email with the text from 2 pdf files as Body and attached with “ScannedDoc.pdf”

Algorithm:

Step 1: START

Step 2: The first PDF document we are trying to extract text from is a native pdf, (the text is selectable, the doc is created from a Word document probably, etc) so we will use the dedicated activity for this case, use **Read PDF Text activity**.

Step 3: Set File Name property and browse to the location we have the Orchestrator Installation Guide document saved at.

Step 4: Replace the Range property value from “All” to 6, in order to read just the page 6 from the document.

Step 5: Create a variable “installationPDFText” in the Output, to save the retrieved text for later use.

Step 6: The second pdf document we are trying to extract from is a scanned pdf (everything is a picture, you cannot select any element, etc) so we will use the dedicated activity for this case, use **Read PDF With OCR** activity.

Step 7: Set File Name property and browse to the location we have the “Scanned.pdf” document.

Step 8: Replace the Range property value from “All” to 2, in order to read just the page 2 from the document. (Adjust the property to ensure that the content is extracted correctly from the pdf file.)

Step 9: Create a variable “invoicePDFText” in the Output, to save the retrieved text for later use.

Step 10: Add a **Send SMTP Mail Message** activity (can be any Send XXX Mail Message activity)set the To property.

- Set the Host server property as you choose i.e "smtp.gmail.com"

- Set the Logon property as you choose i.e. aaa@gmail.com

Step 11: Set the To textbox as your email address (not gmail address)

Step 12: Set the Subject textbox as "**Email automation by UiPath**"

Step 13: Set the Body textbox as the concatenated variables that keep the text pieces read from the two documents installationPDFText + invoicePDFText.

Step 14: Press Attach Files to attach a file and put Environment.CurrentDirectory+" ScannedDoc.pdf with path directory" in the Value.

Outcome:

- An email having "Email automation by UiPath" in Subject and text from 2 pdf files in Body and attached with "ScannedDoc.pdf" file